

# EPIPAINTOR

par Fast Thinking

## Cahier des charges



MAHRAOUI Adam - HAMDOUN Zaky - VARLIETTE Robin - AZIAKA Christian

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Le projet : Epipaintor . . . . .	4
1.2	Le groupe : Fast-thinking . . . . .	4
1.3	Présentation des membres . . . . .	4
1.3.1	Adam Mahraoui . . . . .	4
1.3.2	Christian Aziaka . . . . .	4
1.3.3	Zaky Hamdoun . . . . .	4
1.3.4	Robin Varliette . . . . .	5
<b>2</b>	<b>Le projet</b>	<b>5</b>
2.1	Origines . . . . .	5
2.1.1	Idée . . . . .	5
2.1.2	Nom . . . . .	5
2.2	Cibles . . . . .	6
<b>3</b>	<b>Fonctionnalités du projet</b>	<b>6</b>
3.1	Modification des couleurs de l'image . . . . .	6
3.1.1	Palette de couleurs . . . . .	6
3.1.2	Pipette . . . . .	6
3.1.3	Pinceau . . . . .	7
3.1.4	Seau . . . . .	7
3.1.5	Zone de texte . . . . .	7
3.1.6	Forme 2D . . . . .	7
3.2	Édition de l'image . . . . .	7
3.2.1	Modifier les dimensions . . . . .	7
3.2.2	Effectuer une rotation . . . . .	7
3.2.3	Rogner . . . . .	7
3.2.4	Appliquer des filtres . . . . .	8
<b>4</b>	<b>Etat de l'art</b>	<b>8</b>
4.1	La retouche avant le numérique . . . . .	8
4.2	L'arrivé de paint . . . . .	8
4.3	GIMP un programme "open-source" . . . . .	8
4.4	L'évolution perpétuelle de Photoshop . . . . .	9
<b>5</b>	<b>Aspect algorithmique</b>	<b>9</b>
5.1	Piles et historique . . . . .	9

5.2	Filtres . . . . .	9
5.3	Coloration et remplissage d'image . . . . .	10
5.4	Manipulation d'images . . . . .	10
5.5	Compression d'images . . . . .	10
<b>6</b>	<b>Aspect graphique</b>	<b>10</b>
6.1	Technologies . . . . .	10
6.2	Charte graphique . . . . .	10
<b>7</b>	<b>Déroulement</b>	<b>12</b>
7.1	Répartitions des tâches . . . . .	12
7.2	Planification . . . . .	13
<b>8</b>	<b>Conclusion</b>	<b>13</b>

# 1 Introduction

## 1.1 Le projet : Epipaintor

Notre projet est simple : un photoshop intelligent, permettant d'aider les personnes étudiant ou travaillant à EPITA !

Derrière le doux nom d'**Epipaintor** résidera un utilitaire, permettant de dessiner ou modifier une image.

L'idée est d'implémenter tout un éventail de fonctionnalités graphique, utilisant chacun leur algorithmes propres, sur le modèle du logiciel Photoshop.

## 1.2 Le groupe : Fast-thinking

Notre groupe se nomme **Fast-Thinking** :

Il tire son nom de la vivacité d'esprit de ses membres, et de sa volonté à développer des projets ingénieux. Fast-Thinking est un groupe forme de 4 étudiants en deuxième année de l'école d'ingénieur EPITA, aux origines diverses (Italien, Algérien, Marocain, Français). La diversité culturelle est une des grandes forces de **Fast-Thinking** lui permettant d'avoir une combinaisons de raisonnements différents les uns des autres.

## 1.3 Présentation des membres

### 1.3.1 Adam Mahraoui

Étudiant a EPITA, je suis passionné par le monde de la programmation et du numérique en général. La curiosité est un trait de personnalité qui me caractérise bien et c'est pour cela que cette idée projet m'a plu. J'ai vu de plus en plus apparaître de nouvelles fonctionnalités incroyables sur les logiciel de montage photo en me demandant toujours comment cela fonctionnait mais je n'ai jamais vraiment penché sur le sujet. Grâce a ce projet je vais pouvoir enfin répondre à mes questions en renforçant mes compétences en C sur un projet concret.

### 1.3.2 Christian Aziaka

Bonjour, je suis Christian Aziaka et je suis passionné d'IoT, programmation, montage vidéo, d'UX, montage photo, de design et de production musicale.

### 1.3.3 Zaky Hamdoun

Passionné par l'informatique et la géopolitique, j'ai rejoint l'EPITA avec plus tard l'envie de devenir ingénieur en informatique spécialisé en cybersécurité, ou en big data et intelligence artificielle. Actuellement Trader pour Bright Future Investment Management, je m'occupe du développement d'algorithme de trading automatique. Cependant, mon penchant pour la géopolitique et les métiers de la défense font également que je suis intéressé par l'Open Source Intelligence (OSINT) et les renseignements, qui se résument très souvent à du traitement d'image. J'aime également les mathématiques et la philosophie.

### 1.3.4 Robin Varliette

Bonjour, et enchanté, à quiconque lisant ces lignes !

Je suis Robin Varliette et je suis le leader du groupe Fast-Thinking.

J'ai 20 ans, et je suis un passionné depuis toujours d'algorithmique, d'informatique, de mathématiques, ainsi que d' Intelligence Artificielle.

Amateur de sport, j'affectionne le volley, le handball, ou encore le badminton.

Ouvert à tous les horizons, j'adore découvrir de nouveaux univers par le voyage, ou bien la musique.

Enfin, j'ai pour hobby de partager mes passions en proposant mon aide en algorithmique ou en programmation aux élèves d'EPITA qui en ont besoin.

## 2 Le projet

### 2.1 Origines

#### 2.1.1 Idée

L'idée du projet était a l'origine de faire une intelligence artificielle : Un Chat-bot capable de converser avec un être humain. Elle était née du fait que les membres du groupe sont particulièrement attirés par les intelligences artificielles, eux-mêmes ayant pu en développer une dans le cadre leur projet effectué au semestre passé.

L'idée du Chat-Bot comme Intelligence Artificielle prends ses sources dans deux projets existants :

- ChatGPT, le chat bot aux performances fulgurantes développée par Open-AI, qui a émergée très récemment en 2022.

- Tay.ai, le chat bot développe par Microsoft et utilisable sur Twitter, qui était une expérience de "Conversational Understanding" (en français : Apprentissage en conversation).

Cependant, pour des problèmes de complexité, et de barème, le projet n'a pas été accepté et à du être radicalement changé.

Le projet de faire un utilitaire sur la base de Photoshop nous a alors été proposé par notre professeur en charge des projets de 4ème semestre, et nous avons décidé de suivre ses recommandations afin de pouvoir proposer un projet qualitatif.

#### 2.1.2 Nom

Pour commencer, il est important de rappeler qu'en tant qu'élèves d'EPITA, nous avons pour **devoir** d'honorer notre merveilleuse école. Qui serions nous sans elle, pas vrai ?

Il existe une règle implicite dans la nomenclature des choses a EPITA : Chaque nom doit avoir pour préfixe '**Epi**' que cela soit les associations, les sites internet, les organisations ou projets reliés a l'école ont presque tous comme préfixe '**Epi**'. Ainsi, nous suivons donc cette convention car nous portons les traditions de l'EPITA dans notre coeur.

Au départ, il était question d'appeler notre ChatBot 'EpiHelp'.

Cependant, nous avons grandi avec le jeu 'Akinator', un ChatBot inventé par l'ingénieur informatique français Arnaud Megret, capable de deviner un personnage en posant des questions à un utilisateur.

Cela étant dit, nous avons décidé de rendre honneur à cette prouesse technologique de l'époque en ajoutant '**nator**' à notre préfixe '**Epi**', formant ainsi le nom '**Epinator**'.

Le projet ayant changé en étant un utilitaire graphique, nous avons légèrement changé le nom du projet en remplaçant le **nator** en **paintor**, donnant **Epipaintor**.

## 2.2 Cibles

Notre projet cible un intervalle de personnes assez large, du dessinateur amateur ou dessinateur expérimenté.

De ce fait, notre interface devra à la fois fournir des fonctionnalités poussées, et à la fois devra être facile d'utilisation.

## 3 Fonctionnalités du projet

Bien que l'envie ne nous en manque pas, il est peu probable que nous puissions implémenter un utilitaire graphique de la complexité des logiciels dont nous nous sommes inspirés.

C'est pourquoi notre utilitaire implémentera des fonctionnalités d'une complexité proportionnelle à la faisabilité quant à leur développement.

Nous pensons notamment à modifications d'image, comme par exemple dessiner à l'aide d'un pinceau ou appliquer des filtres sur l'image. Toutes les modifications appliquées à l'image seront affichées en temps réel dans l'interface graphique. Voici quelques exemples de fonctionnalités de notre projet :

### 3.1 Modification des couleurs de l'image

#### 3.1.1 Palette de couleurs

Afin que l'utilisateur puisse dessiner de la couleur de son choix, nous implémenterons une fenêtre cliquable afin qu'il puisse choisir sa couleur parmi un ensemble de couleur basique. Nous envisageons également d'offrir la possibilité du choix de couleur RGB pour que toutes les couleurs possible soit utilisables.

#### 3.1.2 Pipette

L'outil pipette est un outil permettant de récupérer une couleur présente sur la zone de dessin. Cela permet à l'utilisateur de ne pas avoir à tester toutes les couleurs pour continuer un dessin avec une couleur précise.

Une fois la couleur récupérée, il sera possible d'y accéder depuis la palette de couleur.

### 3.1.3 Pinceau

Le pinceau permettra à l'utilisateur de créer des traits sur la zone de dessin. Lorsque cette option sera sélectionnée, l'utilisateur devra maintenir son clic gauche et déplacer sa souris pour tracer les traits de la manière dont il le désire.

Il sera possible de modifier l'épaisseur du trait du pinceau depuis la barre d'outils.

### 3.1.4 Seau

Le seau sera un simple outil de remplissage de forme. Une fois sélectionnée depuis la barre d'outils, il sera possible de cliquer à l'intérieur d'une forme sur la zone de dessin, et cette zone de dessin se verra remplie de la couleur sélectionnée dans la palette de couleur.

### 3.1.5 Zone de texte

Nous pourrions également ajouter un outil "zone de texte". C'est outil est présent dans presque tous les éditeurs d'images, et est un grand classique. Il serait alors possible de customiser la zone de texte par sa police, sa couleur, en gras, en italique...

### 3.1.6 Forme 2D

Il sera implémenté dans notre utilitaire une fenêtre permettant de choisir une forme géométrique qu'il sera ensuite possible de placer sur la zone de dessin. Le nombre de forme géométriques qui sera implémenté dépendra du temps qu'il nous restera. Chaque forme 2D sera créée grâce à des algorithmes simples, qui dépendront de la position du curseur lors du début du clic de l'utilisateur et de la position du curseur lorsque le clic est relâché.

## 3.2 Édition de l'image

### 3.2.1 Modifier les dimensions

L'utilisateur aura la possibilité de sélectionner une partie de son dessin grâce à un outil dédié. Suite à cela, il pourra s'il le désire agrandir cette partie. La difficulté de cette opération réside dans le fait qu'il faille garder la bonne proportion de l'image, sauf si l'utilisateur en formule le souhait.

### 3.2.2 Effectuer une rotation

Il est possible que l'utilisateur veuille faire tourner son image d'un certain angle. Étant donné que la rotation d'une image fait partie des algorithmes les plus simples à implémenter, cette fonctionnalité sera sûrement une des premières implémentées.

### 3.2.3 Rogner

Supprimer des zones du dessin en changeant les dimensions de l'image est une fonctionnalité simple et pourtant très utile.

Lorsque l'outil "Rogner" sera sélectionné, des traits représentant les dimensions de l'image apparaîtront, et il suffira à l'utilisateur de faire glisser ses nouvelles dimensions afin de rogner l'image.

### 3.2.4 Appliquer des filtres

Nous rappelons que nous souhaitons créer un système de **calques**. Cela implique que nous devons par la même occasion implémenter le concept de **transparence** dans notre logiciel, sans quoi il serait impossible de superposer plusieurs calques.

Il est cependant important de noter que tous les format d'images ne supporte pas le concept de transparence (notamment le format .jpeg). Cependant, le format .png supporte le concept de transparence, utilisant une variable *alpha* correspondant au pourcentage d'opacité d'un pixel.

Concernant nos filtres, leur application sera effective sur l'entièreté des pixels d'un calque. Ils permettront de modifier l'image de plusieurs manières : *Noir&Blanc*, *Sepia*, *Inversion de couleur*, *Contraste*, *Luminosité*, *Saturation*, *Netteté*.

## 4 Etat de l'art

### 4.1 La retouche avant le numérique

Bien avant la banalisation de Photoshop et les retouches numériques, certains politiciens imposaient déjà leur volonté de modifier chaque détail à leur avantage. Et à l'époque de l'argentique et du noir et blanc, les retoucheurs tentaient déjà de modifier les photographies avec différents procédés ingénieux. Mais alors, comment s'y prenaient-ils ? Assis devant un pupitre à retoucher, les spécialistes plaçaient le cliché sur une table lumineuse. À l'aide de crayons finement taillés, de pinceaux et même de pigments de couleurs, les retoucheurs travaillaient directement sur le négatif en gélatine. Quelquefois, le positif était aussi retouché et les images en noir et blanc l'étaient directement à l'encre de Chine et à l'aide de fins pinceaux. Les retoucheurs utilisaient alors la technique du pointillage qui consistait à réduire les irrégularités d'une image. Généralement, un service de retouche photo coûtait très cher et seuls les hommes politiques et les personnalités pouvaient se permettre de faire appel à ce type de service. On sait par exemple qu'Abraham Lincoln avait fait retoucher une image de lui dans les années 1860.

### 4.2 L'arrivée de paint

Par la suite en 1985 c'est le fameux logiciel Paint qui est officiellement sorti sur les ordinateurs dotés de Windows. Cette première version permettait uniquement l'utilisation de graphiques monochromes et les sauvegardait. C'est par la suite, avec l'ajout de plusieurs mises à jour où la logicielle a pu augmenter le nombre d'outils à disposition de l'utilisateur comme par exemple la possibilité d'extraire des images directement à partir d'un scanner ou d'un appareil photo numérique connecté au PC, utilisation de tampon, modification du contraste.

### 4.3 GIMP un programme "open-source"

Lancé en 1995 par Spencer Kimball et Peter Mattis, GIMP signifiait initialement General Image Manipulation Program. C'est un des premiers logiciels libres de droit dans le domaine de l'édition d'image numérique, en effet le programme fait partie du GNU Project visait à créer et partager un maximum de logiciels libres au public en leur donnant non seulement les droits d'utilisation, mais aussi de modification et distribution de ceux-ci.



GIMP propose de nombreux outils différents que ce soit le crayon, le pinceau ou encore la baguette. Cette dernière permet de sélectionner des parties spécifiques d'une image pour les modifier. On peut facilement réaliser des formes simples tels qu'un cercle ou un rectangle avec les outils de sélection. L'outil texte permet d'écrire sur l'image avec une taille et une police définies.

## 4.4 L'évolution perpétuelle de Photoshop

Photoshop, actuellement, est le plus populaire des logiciels de retouches de l'imagerie digitale. En simple, Photoshop n'est ni plus ni moins qu'un laboratoire numérique de développement, de retouches de photos et d'images. C'est en 1987, Thomas Knoll, un étudiant de l'université du Michigan, commence l'écriture d'un programme pour afficher des images en niveaux de gris sur écran monochrome. Appelé « Display », ce programme attire l'attention de son frère John Knoll, alors employé chez Industrial Light Magic. Ce dernier lui recommande d'en faire un éditeur d'images à part entière qui deviendra des années plus tard le logiciel que nous utilisons aujourd'hui.

Photoshop possède son propre format de fichier (extension psd). Celui-ci permet de conserver distincts les différents calques formant l'image afin de les manipuler séparément.

La modernisation constante du logiciel avec les années a été le point le plus important qui a permis à Photoshop de devenir ce qui l'est aujourd'hui. D'un point de vue fonctionnel l'application possède une large gamme d'outils très sophistiqués qui permet aux utilisateurs de toujours pousser la création plus loin. Un de ses principaux avantages avec les logiciels concurrents est sa constante évolution. En effet, avec l'arrivée massive des IA, le logiciel s'est vu inclure des fonctionnalités très poussées telles que la suppression d'élément automatisé qui prenait un temps important auparavant. Il permet aussi de retoucher de manière automatisée et optimale une photo ou la colorisation de photo en noir et blanc. La modernisation constante du logiciel avec les années a été le point le plus important qui a permis à Photoshop de devenir ce qui l'est aujourd'hui.

## 5 Aspect algorithmique

### 5.1 Piles et historique

Comme dans tout éditeur d'image, il existera une fonctionnalité historique, qui permettra de retourner en arrière et d'annuler certaines modifications dans un certain ordre. Ce concept d'historique et d'annulation se base spécifiquement sur le fonctionnement des piles, une base de données du type LIFO (Last-In First-Out), il faudra donc implémenter cette structure de données.

### 5.2 Filtres

Dans un premier temps, il est important de noter le fait que les filtres qui seront implantés sont tous basés sur des concepts algorithmiques bien précis. En l'occurrence, l'aspect algorithmique sera ici basé sur les différentes manipulations des pixels, et, par conséquent, la traversée des matrices.

## 5.3 Coloration et remplissage d'image

Plusieurs algorithmes de coloration d'image existent. L'algorithme de remplissage est un algorithme très important qui sera notamment utiliser dans le but de la coloration et du remplacement de pixels de la même couleur. L'algorithme baguette magique utilisera notamment les principes du filling récursif.

## 5.4 Manipulation d'images

Les fonctionnalités de base, à savoir, la rotation manuelle et la rotation automatique nécessitent également une certaine connaissance algorithmique. Ces deux fonctionnalités seront aussi implémentées. Toutefois, cette partie suppose aussi la création des différents algorithmes de redimensionnement d'images, qui sont également essentielles à tout éditeur d'image de qualité.

## 5.5 Compression d'images

Les éditeurs d'images possèdent souvent des systèmes de compression d'images. La compression d'image est une application de la compression de données sur des images numériques. Cette compression a pour utilité de réduire la redondance des données d'une image afin de pouvoir l'emmagasiner sans occuper beaucoup d'espace ou la transmettre rapidement.

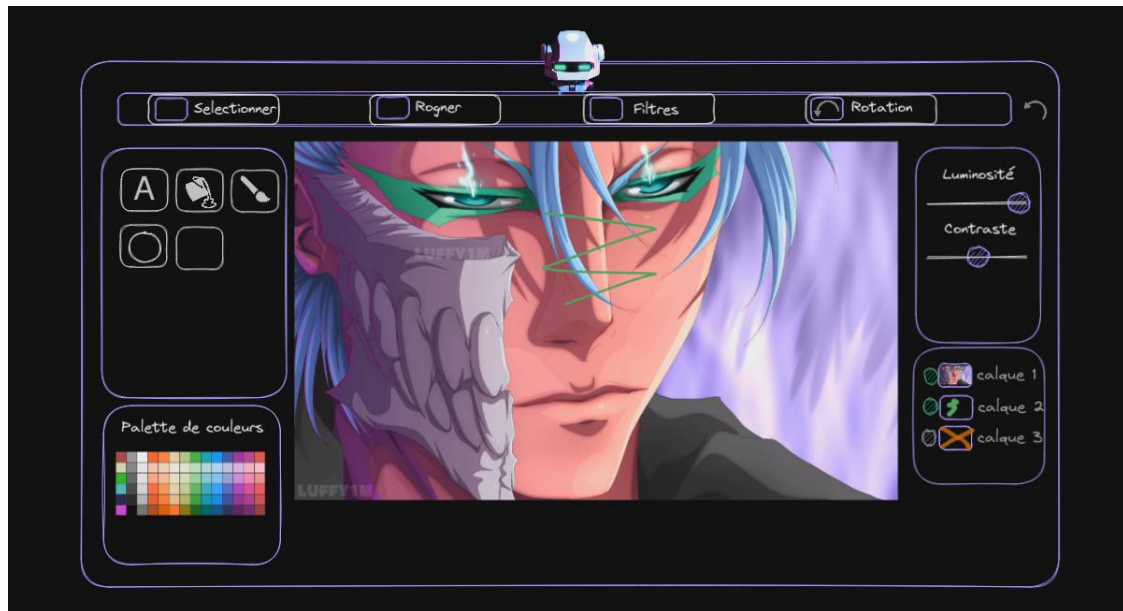
# 6 Aspect graphique

## 6.1 Technologies

Les logiciels que nous utiliserons pour traiter la partie graphique sont GTK et GIMP. GTK va nous servir à implémenter nos fonctionnalités rendre notre interface dynamique. GIMP va nous servir ça personnaliser notre interface et à personnaliser la disposition de ses composants.

## 6.2 Charte graphique

Notre couleur dominante pour notre interface est le violet. Voici un exemple de notre interface graphique :



Sur la partie supérieure de notre interface nous pouvons retrouver la partie "Édition de l'image" avec l'outil sélection, rogner, filtres, rotation. Sur la partie droite nous avons les réglages de luminosité de contraste et le système de calque : si le bouton est vert cela veut dire que le calque est visible sinon il est "caché". Sur la droite nous avons la partie "Modification des couleurs de l'image" avec l'outil pinceau, remplissage, écriture de texte et sélection de formes

## 7 Déroulement

### 7.1 Répartitions des taches

Tâches	Adam	Zaky	Christian	Robin
<b>SDL</b>				
Gestion des images			R	S
Gestion des couleurs		S	R	
Gestion de la structure des images	S	R		
Filtres			S	R
<b>GTK - Interface</b>				
Menu utilisateur	S			R
Développement des outils		S		R
Ouverture / Sauvegarde de fichiers			R	S
Zone de dessin	R	S		
Pop-ups			R	S
<b>Structure</b>				
Design de l'application		S	R	
Architecture des fichiers sources	S			R
Raccord des parties (Compilation, makefile)	R	S		
<b>Présentation</b>				
Site internet			R	
Documents latex	S	R		

TABLE 1 – Répartition des Tâches - S :Suppléant - R : Responsable

## 7.2 Planification

Tâches	Soutenance 1	Soutenance 2	Soutenance 3
<b>GTK - Interface</b>			
Menu utilisateur	100%	100%	100%
Développement des outils	50%	75%	100%
Ouverture / Sauvegarde de fichiers	100%	100%	100%
Zone de dessin	50%	80%	100%
Pop-ups	45%	100%	100%
<b>SDL</b>			
Gestion des images	50%	85%	100%
Gestion des couleurs	70%	95%	100%
Gestion de la structure des images	70%	100%	100%
Filtres	50%	750%	100%
<b>Interface graphique</b>			
Design de l'application	40%	70%	100%
Architecture des fichiers sources	70%	100%	100%
Raccord des parties	30%	60%	100%

TABLE 2 – Planification de l'avancement des tâches

## 8 Conclusion

Pour conclure, ce projet va permettre à chaque membre du groupe de renforcer ses compétences avec le langage C en participant à la réalisation d'un projet concret et immersif. Cela nous permettra aussi de renforcer notre méthode de travail du groupe ce qui est devenu primordial dans le monde du travail d'aujourd'hui.