

EPITA

---

# Rapport de soutenance 2 GMPF

---

par *void*\*

Samory DIABY

Pierrick GUILLAUME

Arnaud CASSEZ

Sabine HU

03 avril 2019

# Table des matières

<b>Introduction</b>	<b>3</b>
<b>1 Retour rapide sur le cahier des charges</b>	<b>4</b>
1.1 Planning . . . . .	4
1.2 Répartition des tâches . . . . .	5
<b>2 Présentations</b>	<b>6</b>
2.1 Présentations individuelles . . . . .	6
2.1.1 Samory DIABY . . . . .	6
2.1.2 Pierrick GUILLAUME . . . . .	6
2.1.3 Arnaud CASSEZ . . . . .	7
2.1.4 Sabine HU . . . . .	7
2.2 Présentation du projet . . . . .	8
<b>3 Avancement du projet</b>	<b>9</b>
3.1 Transformation de l'image . . . . .	9
3.2 Interface . . . . .	10
3.3 Gestion des calques . . . . .	13
3.3.1 Gestionnaire des calques . . . . .	13
3.3.2 Calques . . . . .	13
3.4 Sauvegarde des calques . . . . .	14
3.4.1 Exportation des calques . . . . .	14
3.4.2 Sauvegarde des projets . . . . .	15
3.5 Filtres . . . . .	15
3.6 Créeateur de filtres . . . . .	21
3.7 Exemples d'images . . . . .	22
<b>4 Outils</b>	<b>23</b>
4.1 Taille du curseur ajustable . . . . .	23
4.2 Pinceau . . . . .	23
4.3 Gomme . . . . .	24

---

<b>5</b>	<b>Site internet</b>	<b>25</b>
5.1	Intérêt . . . . .	25
5.2	Structure du site . . . . .	25
<b>6</b>	<b>Réalisations de la première soutenance</b>	<b>27</b>
<b>7</b>	<b>Réalisation de la seconde soutenance</b>	<b>27</b>
<b>8</b>	<b>Objectifs pour la prochaine soutenance</b>	<b>29</b>
	<b>Conclusion</b>	<b>30</b>

## Introduction

Nous nous sommes fixés pour objectif de réaliser le projet *GMPF* sur une période de cinq mois. Nous voici arrivés à la deuxième soutenance et déjà quatre mois se sont écoulés et nous avons chacun contribué à l'avancement du projet. Nous sommes pour le moment légèrement en avance par rapport à nos objectifs de temps fixé sur certaines parties et dans les temps pour les autres. Il s'agit lors de cette soutenance de présenter le travail qui a été réalisé. Nous retrouverons également les difficultés rencontrées ainsi que les éventuelles améliorations à apporter aux éléments déjà réalisés.

Durant ces quatre premiers mois, nous avons eu besoin de faire preuve de rigueur ainsi que d'écoute envers chacun des membres afin de pouvoir s'aider mutuellement et ainsi, avancer de manière efficace dans la réalisation du projet.

# 1 Retour rapide sur le cahier des charges

## 1.1 Planning

	1 <sup>ère</sup> Soutenance	2 <sup>ème</sup> Soutenance	Soutenance Finale
<b>PROJET</b>			
<b>Calques</b>			
Gestion de calques	50%	80%	100%
Rotation de calques/images	50%	100%	100%
Re-dimension de calques/images	50%	100%	100%
<b>Filtres</b>			
Créateur de filtres	50%	80%	100%
Filtres pré-installés	70%	100%	100%
<b>Outils</b>			
Taille du curseur ajustable	20%	50%	100%
Effets	20%	70%	100%
<b>Sauvegarde</b>			
Sauvegarde de calques	0%	50%	100%
Sauvegarde du projet (ensemble de calques)	0%	25%	100%
<b>Interface graphique</b>			
Interface	25%	60%	100%
<b>BONUS</b>			
Possibilité de dessiner		-	
<b>SITE INTERNET</b>			
Interface	33%	67%	100%
Téléchargement du projet	0%	50%	100%

## 1.2 Répartition des tâches

	Samory	Pierrick	Arnaud	Sabine
<b>PROJET</b>				
<b>Calques</b>				
Gestion de calques	O		X	
Rotation de calques/images		O	X	
Re-dimension de calques/images			X	O
<b>Filtres</b>				
Créateur de filtres	X			O
Filtres pré-installés			O	X
<b>Outils</b>				
Taille du curseur ajustable	X	O		
Effets		O		X
<b>Sauvegarde</b>				
Sauvegarde de calques	X		O	
Sauvegarde du projet (ensemble de calques)	O		X	
<b>Interface graphique</b>				
Interface	O	X		
<b>BONUS</b>				
Possibilité de dessiner			-	
<b>SITE INTERNET</b>				
Interface			O	X
Téléchargement du projet		X		O

### LÉGENDE

X : responsable

O : suppléant

## 2 Présentations

### 2.1 Présentations individuelles

#### 2.1.1 Samory DIABY

Je m'appelle Samory Diaby. Je m'intéresse beaucoup au domaine de la robotique que j'apprécie particulièrement, et qui m'a poussé à construire quelques montages et petits robots à l'aide de la plate-forme *arduino*. Je pratique également différentes activités sportives comme le tennis ou le basket-ball, car selon moi, il est important de toujours pratiquer une activité physique.

#### 2.1.2 Pierrick GUILLAUME

Je suis cette année étudiant à EPITA, j'espère réaliser durant ce quatrième semestre mon dernier projet en cycle préparatoire. L'année dernière, j'ai eu la chance de faire un projet de groupe. Nous devions réaliser un jeu Snake en trois dimensions, grâce à l'outil Unity. Cette année, l'objectif sera de créer une application de retouche d'images similaire à *Photoshop* ou *GIMP* mais en un peu plus simple. En effet, ces logiciels ont été développés par des professionnels et sur plusieurs années.

J'espère réussir à travailler de manière efficace avec les membres de mon groupe afin de travailler main dans la main vers un but commun et aboutir à une application efficace et simple d'utilisation.

Le projet de l'année dernière m'a montré comment il pouvait être difficile de coordonner les efforts dans un groupe pour obtenir un projet respectant au mieux le cahier des charges initiale. Je m'étais retrouvé chef de projet et il avait fallu que je motive les membres de mon groupe à plusieurs reprise au cours de l'année.

### 2.1.3 Arnaud CASSEZ

Je m'appelle Arnaud Cassez. Avant d'entrer à Epita, je n'avais jamais vraiment codé. J'ai pu, l'an dernier, progresser grâce au projet et aux cours de programmation. Cette année, le projet est de bien plus grande envergure et s'annonce comme un défi encore plus grand que l'an passé et je compte donner le meilleur de moi-même pour aider le groupe.

L'idée de faire un logiciel d'édition d'images m'inspire beaucoup. J'espère en apprendre encore plus sur le langage *C* au cours de ce projet et renforcer les connaissances acquises lors des cours de programmation.

### 2.1.4 Sabine HU

Durant ces derniers mois, j'ai eu l'occasion de réaliser deux projets en groupe : un jeu vidéo ainsi qu'un logiciel de reconnaissance de caractères. Grâce à ces deux projets, j'ai acquis de nombreuses connaissances qui me sont très utiles. En effet, lors de la réalisation du jeu, j'ai pu découvrir comment réaliser un site et lors du second projet, j'ai implémenté un réseau de neurones capable de reconnaître des caractères.

Le projet GMPF, un logiciel de retouche d'images, est une fois de plus une occasion à saisir pour travailler en groupe. Ayant déjà réalisé un projet avec certains coéquipiers, j'ai entièrement confiance en eux pour réaliser à au mieux ce nouveau projet. De mon côté, j'espère acquérir de nouvelles connaissances durant ces prochains mois.

## 2.2 Présentation du projet

Le groupe *void\** est composé de :

Samory DIABY	(samory.diaby) - Chef de projet
Sabine HU	(sabine.hu)
Pierrick GUILLAUME	(pierrick.guillaume)
Arnaud CASSEZ	(arnaud.cassez)

Le projet s'appelle *GMPF* pour *Graphic Maestro Photo Formaat*. Comme son nom l'indique, nous avons choisi de nous orienter vers un logiciel d'édition et de retouche d'images. Le nom du projet est composé de mots issus de différentes langues (anglais, espagnol, français et néerlandais) pour mettre en avant le fait qu'il est polyvalent et qu'il permet de faire beaucoup d'actions sur les images.

### Description du projet

Notre projet est, comme dit précédemment, un logiciel de retouche et d'édition d'images. Il permettra donc d'interagir avec des photos grâce à l'application de filtres, de fusion d'images, de rotations sur ces dernières, etc. Le principe du projet est de pouvoir facilement et intuitivement apporter des modifications à ses images grâce aux outils que nous fournirons avec une interface graphique.

Il permettra donc de charger des images et de les sauvegarder, ainsi que sauvegarder les modifications intermédiaires dans des fichiers spécifiques à notre application pour pouvoir reprendre la modification plus tard.

### 3 Avancement du projet

#### 3.1 Transformation de l'image

Dans l'application *GMPF*, il est possible d'effectuer des transformations sur l'image telles que la rotation, l'inversion, ainsi que le re-dimensionnement de l'image. La partie rotation a été effectuée à l'aide de la bibliothèque *Cairo*. En effet, celle-ci propose des outils qui simplifient la rotation des images en passant par des "*surface de dessin*" : ce sont des éléments propres à cette bibliothèque qui utilise des matrices de pixels afin d'améliorer les performances lors de la modification des images puisqu'il s'agit seulement d'appliquer des transformations à la matrice contenant les pixels. Cette bibliothèque propose des opérations relativement simple en théorie, mais nous avons également dû faire preuve d'imagination pour trouver le système que nous allons utiliser afin de pouvoir modifier chaque calque individuellement.

Il y a ensuite le re-dimensionnement de l'image. Cette partie a pris un peu plus de temps car *Cairo* fournit une fonction permettant de re-dimensionner une surface avec l'application d'opérations sur les matrices. Cependant, cette partie nous a posé beaucoup de problèmes car il a fallu penser à une manière d'appliquer ces changements aux différents calques déjà présents dans un projet. Il a donc fallu faire preuve de réflexion nous permettant ainsi de trouver une solution qui convenait à tous. Cette dernière a donc été mise en place et fonctionne comme nous le souhaitons. Notre solution permet de redimensionner l'ensemble des calques afin de permettre à l'utilisateur d'agrandir l'image d'afin d'effectuer un travail avec plus de précision. L'utilisateur interagit sur le facteur de grossissement par l'intermédiaire d'un menu déroulant ainsi qu'une zone de saisie afin d'entrer le facteur souhaité.

### 3.2 Interface



La première ligne en haut de la fenêtre nous permet d'afficher les différents menu de notre application afin de permettre à l'utilisateur d'accéder à toutes les fonctionnalités d'un simple clic. Elles sont organisées par catégories afin que l'utilisateur puisse retrouver facilement ce qu'il souhaite.

Comme on peut le voir sur l'image, il est possible de choisir de facteur de grossissement souhaité directement depuis un menu déroulant. Il est également possible d'entrer soi-même le facteur désiré afin de donner un meilleur contrôle à l'utilisateur.

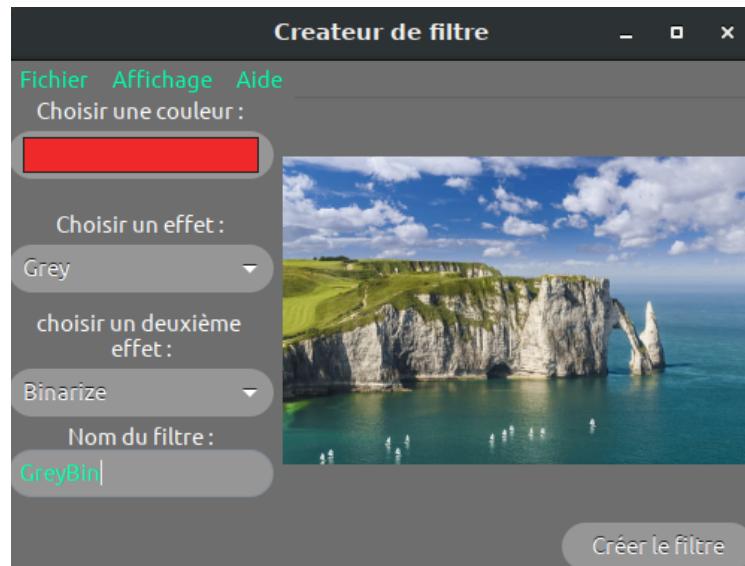
Au centre de la fenêtre, nous avons décidé d'afficher l'aperçu de l'image sur laquelle l'utilisateur est en train de travailler. Elle est affichée dans une fenêtre défilante afin de ne pas avoir une fenêtre immense lorsque l'utilisateur souhaite avoir un aperçu plus rapproché de l'image. Cela nous permet de faire un grossissement fois cent sur l'image tout en la visualisant dans son intégralité.

La partie droite de la fenêtre nous permet d'afficher les différents calques appliqués à notre image. Nous avons donc un aperçu des différentes couches successives. L'utilisateur a donc la possibilité d'en retirer si elles ne présentent plus aucune utilité. Celles-ci sont représentées sous la forme de liste

de calques de haut en bas. Il est désormais possible d’interagir sur l’image directement à l’aide des calques puisque l’utilisateur a la possibilité de les ordonner dans l’ordre qu’il souhaite. Les calques s’affichant les uns par dessus les autres, cela modifie l’image qui se trouvera au premier plan.

Il est possible de voir sur la gauche une petite bande avec plusieurs boutons. Ils représentent les différents outils mis à disposition pour interagir avec l’image. Le premier outil est un pinceau. Il permet donc de dessiner avec une largeur donnée et la couleur choisie par l’utilisateur dans la barre verticale située au-dessus. Le second outil est une gomme. Il reprend le même principe que le pinceau mais celui-ci remplace les pixels colorés par des pixels transparents. Le troisième est une pipette à couleur. Cet outil n’est pas encore complètement implémenté mais servira par la suite à sélectionner une couleur déjà présente sur l’image. Enfin, le dernier est un outil de sélection rectangulaire. Il permet de sélectionner une partie de l’image et devrait par la suite permettre la modification de la partie sélectionnée.

Enfin, la ligne du bas affiche le nom de l’image sur laquelle l’utilisateur travaille. C’est également un bouton qui lui permet d’aller chercher une autre image sur son ordinateur pour la charger dans l’application.



Il existe également un fenêtre dédié à la création d'un filtre qui est appelé lorsque l'utilisateur souhaite en créer un nouveau. Il y a dans cette fenêtre un menu déroulant permettant de choisir une couleur à appliquer, mais également deux autres menus déroulant permettant de sélectionner deux filtres afin de les combiner en un seul. Un fichier texte du nom donné au filtre est créé lorsque l'utilisateur appuie sur le bouton *Créer le filtre*. Ce filtre est alors réutilisable et peut être rechargé depuis l'interface.

### 3.3 Gestion des calques

Pour cette deuxième soutenance, nous avons continué d'implémenter le gestionnaire de calques, permettant ainsi à l'utilisateur de modifier plusieurs images séparément avec différents effets.

#### 3.3.1 Gestionnaire des calques

Pour la première soutenance, le gestionnaire de calques, que nous avons décidé de nommer "GM<sup>P</sup>F\_LayerMngr" pour "Manager de Calques de GM<sup>P</sup>F", permettait de créer et de supprimer des calques. Ceux-ci apparaissaient ensuite dans la partie de la fenêtre que nous lui avons réservée. L'utilisateur pouvait donc sélectionner un calque, même si nous n'avions pas encore relié le gestionnaire avec la partie affichage de l'image.

Après modifications, ce gestionnaire de calques nous permet à présent de modifier l'ordre des calques, et donc le rendu de l'image. Nous pouvons également avoir un aperçu des images présentes sur chaque calque lorsque celui-ci est chargée dans *GM<sup>P</sup>F*.

#### 3.3.2 Calques

Nous avons aussi implémenté des *GM<sup>P</sup>F\_Layer*. Ces derniers correspondent tout simplement aux calques. Nous avons décidé de les lier par une liste chaînée, avec en sentinelle le *GM<sup>P</sup>F\_LayerMngr*. Pour sa part, le calque ne contient qu'une image accompagnée d'une position, d'une taille, d'un booleen qui définit sa visibilité et d'un pointeur sur l'objet qui le représente dans l'interface pour pouvoir le mettre à jour le plus rapidement possible. On ne peut pour le moment choisir la position du calque sur l'image de finale que lors de l'initialisation du calque, mais nous devrions pouvoir à terme déplacer les calques lors de leur utilisation dans le programme.

Par rapport à la dernière soutenance, nous avons décidé de passer sur la gestion des images par la bibliothèque *cairo*. Cette bibliothèque est bien plus puissante que *Gtk* pour la gestion des images. En effet, *Cairo* utilise des matrices de pixels afin d'appliquer des opérations plus ou moins complexe en un temps record.

Cette bibliothèque nous permet également de dessiner sur l'image, une fonctionnalité bonus mais nous nous sommes dit que ce serait dommage de créer un éditeur d'image sans avoir la possibilité de dessiner dessus. Nous avons donc dû adapter le travail déjà effectué, c'est-à-dire la gestion de l'image et l'application des filtres. Nous avons profité de cette refonte pour ajouter une bien meilleur intégration des calques à notre image. En effet, l'image final est composée de plusieurs calques affichés les uns par dessus les autres avec chacun un angle de rotation distinct, ainsi qu'une position et une taille propre, ce qui nous permet de les positionner où l'on souhaite sur l'image, ainsi que de définir la taille de ceux-ci, définissant alors la zone désirable de chaque calque sur l'image.

## 3.4 Sauvegarde des calques

### 3.4.1 Exportation des calques

Nous avons également ajouté la fonctionnalité d'exportation de l'image vers un format d'image *png*. La fonction d'exportation va récupérer chaque *cairo surface* présent sur les calques, afin de les recréer sur une *cairo surface* temporaire que nous transformons en *png* par la suite. L'image est ensuite sauvegardée avec le nom entré plus tôt par l'utilisateur.

Notre avancement sur le gestionnaire de calques nous semble satisfaisant. Nous devrons cependant concentrer nos efforts sur celui-ci pour la prochaine soutenance afin que son intégration au projet soit la meilleure possible. Nous allons également devoir relier les différents filtres et effets avec le gestionnaire de calques pour pouvoir appliquer les filtres uniquement sur les

calques sélectionnés par l'utilisateur.

### 3.4.2 Sauvegarde des projets

Pour réaliser la sauvegarde du projet, nous nous sommes servis de la bibliothèque standard *stdio.h*. Nous utilisons pour cela les fonctions *fread* et *fwrite* qui permettent d'écrire en binaire dans un fichier.

La sauvegarde se fait par plusieurs étapes successives : d'abord nous sauvegardons le *GMPF LayerMngr* sans prendre en compte les cas particuliers des pointeurs (les pointeurs deviennent faux en dehors de leur contexte). Nous sauvegardons ensuite les *GMPF Layer* un par un avec en premier lieu celle du calque de la même manière que pour le gestionnaire de calques, puis nous rajoutons la taille de l'image liée au calque avant de sauvegarder cette dernière.

Pour ce qui est de la restauration / chargement du projet, nous chargeons les éléments dans l'ordre de sauvegarde. Nous devons, durant le chargement, recréer les calques et le gestionnaire de calques et c'est à ce moment là que nous ré-attribuons les pointeurs de ces structures.

## 3.5 Filtres

Pour cette deuxième soutenance, nous avons ré-implémenté les filtres déjà présents lors de la première soutenance, tout en les améliorant afin qu'ils soient compatible avec notre nouvelle implémentation des calques.

Lors de la première soutenance, les filtres que nous avions implémentés sont les suivants : *Tinter*, *Grayscale*, *Binarize*, *Binarize color*, *Negative*, *Blur*, *Enhance*, *Sharpen*. Notons que les trois derniers filtres cités ci-dessus n'étaient pas encore reliés à l'interface. En effet, nous avions rencontré quelques problèmes qui sont désormais résolus.

Dans cette partie, nous allons présenter les filtres qui ont été implémentés depuis la précédente soutenance. Notons que chacun de ces filtres sera suivi d'une image correspond à l'application du filtre. L'image d'origine est la suivante.



FIGURE 1 – Image d'origine

## Chaud

Le filtre *chaud*, également appelé *warm*, nous permet d'obtenir une image dont les couleurs sont plus chaudes. En effet, chaque valeur RGB des pixels est augmentée d'un certain pourcentage. Il s'agit principalement de la valeur rouge des pixels qui admet une augmentation plus importante que les autres. Nous obtenons ainsi l'image suivante.

FIGURE 2 – Filtre *Chaud*

## Froid

Le filtre *froid*, également appelé *cold*, nous permet d’obtenir une image dont les couleurs sont plus froides. Ce filtre s’oppose ainsi au filtre précédent, *Warm*. En effet, ce filtre s’intéresse principalement à la valeur bleue des pixels. Après application de ce filtre, nous obtenons ainsi l’image suivante.

FIGURE 3 – Filtre *Froid*

## Vert

Le filtre *vert* s'inspire des deux filtres précédents. En effet, l'un d'eux se concentrat principalement sur la composante rouge et l'autre sur la composante bleue. Le filtre *vert* s'appuie donc sur la composante verte de chaque pixel. Ci-contre, nous pouvons retrouver l'image obtenue.



FIGURE 4 – Filtre *Vert*

## Luminosité +

Le filtre *luminosité +* fait apparaître l'image plus clair. En effet, chaque valeur RGB des pixels est augmentée de 25 afin de se rapprocher davantage de 255, la valeur correspondante à la couleur blanche.

FIGURE 5 – Filtre *Luminosité +*

### Luminosité -

Le filtre *luminosité -* fait apparaître l'image plus sombre. En effet, contrairement au précédent filtre, celui-ci récupère chaque valeur RGB des pixels et soustrait à ces dernières la valeur 25. Nous souhaitons ainsi nous rapprocher de la couleur noire dont les valeurs RGB sont à 0.

FIGURE 6 – Filtre *Luminosité -*

## Égalisation en niveau de gris

Le filtre *égalisation*, également appelé *equalization* en anglais, nous renvoie une image en niveau de gris. Contrairement au filtre *Grayscale* que nous avons implémenté pour la précédente soutenance, ce dernier distingue davantage les couleurs. En effet, ce filtre nécessite l'utilisation des histogrammes de taille 255. Pour chaque valeur de gris obtenue lorsque de l'application du *Grayscale*, nous incrémentons la valeur correspondante dans l'histogramme. Dans l'histogramme, nous appliquons la fonction ci-dessous à chacune des valeurs et le résultat obtenu est la valeur grise que nous appliquons à l'image.

$$h(v) = \text{round} \left( \frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L - 1) \right)$$

*Round* renvoie un entier à partir d'un nombre décimal.

*Cdf* correspond au nombre d'occurrence de la valeur v grise du *Grayscale*.

*L* correspond au nombre maximal de valeur de gris. Elle est donc égale à 255.

*M* et *N* correspondent respectivement à la longueur et à la largeur de l'image.



FIGURE 7 – Filtre *Egalisation en niveau de gris*

## Égalisation en couleurs

Le filtre *égalisation en couleurs* possède le même principe que le filtre précédent. Ce dernier effectue les mêmes calculs mais pour chaque valeur rouge, verte et bleue de chaque pixel. A l'issu de l'application du filtre, nous obtenons l'image suivante.



FIGURE 8 – Filtre *Egalisation en couleurs*

## 3.6 Créeur de filtres

Pour cette deuxième soutenance, nous avons continuer d'implémenter le créateur de filtre afin que celui puisse correspondre au mieux à ce que nous avons indiqué dans le cahier des charges. Nous avons donc pour cela ajouter la possibilité d'ajouter une 2 ème filtre afin d'augmenter le nombre de combinaisons possible, nous souhaitons cependant ajouter la possibilité de choisir entre 3 ou 4 filtres afin d'augmenter davantage le nombre de combinaisons.

Après avoir sélectionner des filtres et une couleur à appliquer à ceux-ci, l'utilisateur aura désormais la possibilité de sauvegarder son filtre avec le nom qu'il souhaite lui donner. Le filtre personnalisé pourra donc être rechargé depuis le menu *filtre* de notre logiciel, et être appliqué directement sur le calque sélectionné.

### 3.7 Exemples d'images

Ci-dessous, nous pouvons retrouver deux images qui ont été retouchées par notre propre logiciel. Cela nous permet donc de voir l'avancement du projet.

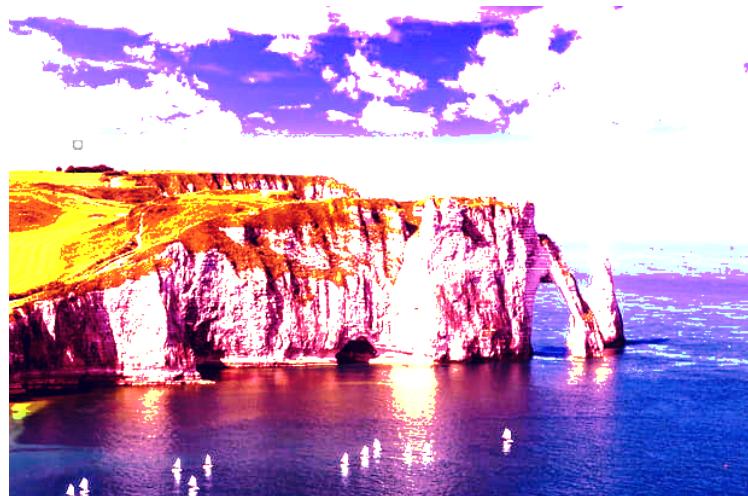


FIGURE 9 – Image retouchée 1



FIGURE 10 – Image retouchée 2

## 4 Outils

La partie *Outils* de notre projet consiste à la création de différents effets qui viendront par la suite s'ajouter aux filtres que nous avons implémentés. Nous avons déjà ajouté sur l'interface différents boutons permettant de dessiner et de gommer sur le calque, cependant il nous reste encore à implémenter les différentes fonctions reliées aux boutons restants, et qui serviront à découper une zone du calque à l'aide de l'outil *sélection* et à sélectionner une couleur du calque à l'aide de l'outil *pipette*.

### 4.1 Taille du curseur ajustable

Pour cette deuxième soutenance, nous avons avancé dans l'implémentation du *curseur ajustable*. Nous avons donc la possibilité de changer la taille curseur lorsque nous nous déplaçons sur la zone de modifications de calque. Actuellement le curseur prend la forme d'un cercle à l'intérieur de la zone de modifications, et revient à la normale lorsque le curseur sort de la zone.

Ce curseur permet actuellement de dessiner à l'aide du pinceau sur le calque sélectionné, et également de gommer à l'aide de l'outil *gomme* à la taille souhaitée par l'utilisateur.

### 4.2 Pinceau

Pour cette deuxième soutenance, nous avons pu implémenter différents bonus, tels que le pinceau et la gomme, qui sont liés à la sous partie précédente.

L'outil pinceau permet à l'utilisateur de dessiner sur le calque qu'il a sélectionné à l'aide du gestionnaire de calques. Il pourra donc choisir une couleur avec le bouton de sélection de couleur *GTK* et choisir une taille de pinceau comme expliqué précédemment. Cependant, nous avons rencontré un problème lors de la création du pinceau. En effet, ce dernier ne peignait pas

continuellement à cause de la compression des événements de *GTK*. Après modifications des fonctions, nous avons finalement réussi à supprimer cette compression des évènements et à dessiner une ligne entre deux points de l'écran, ce qui nous permet donc de dessiner de manière continue.

### 4.3 Gomme

L'outil gomme possède un fonctionnement similaire à celui du pinceau, à l'exception près que la gomme peint avec une valeur *alpha* nulle, ce qui lui permet "d'effacer" les pixels du calque sélectionné. Il est également possible de re-dimensionner la gomme afin de gommer un nombre plus ou moins important de pixels.

## 5 Site internet

### 5.1 Intérêt

Étant donné que la réalisation d'un site est obligatoire pour ce projet, nous avons réalisé un site épuré et facile d'accès. Nous pouvons y retrouver diverses rubriques comme un onglet de présentation, une zone d'actualités et une page pour le téléchargement du projet. L'onglet de présentation regroupera les informations relatives au groupe et au projet.

Sur la page d'accueil, nous retrouverons bien évidemment une brève présentation du projet. Sur la page de actualités, nous pouvons accéder à l'historique correspondant à l'avancée du projet ainsi que les problèmes rencontrés et leur solutions.

Dans l'onglet de téléchargement, nous pourrons bien évidemment télécharger le logiciel, mais aussi accéder aux différents rapports de soutenances et au cahier des charges. De plus, nous y ajouterons les liens des sites qui nous ont permis d'avancer le projet (images utilisées, librairies et logiciels).

### 5.2 Structure du site

La réalisation du site est restée assez simpliste. En effet, un site ne doit pas être trop encombrant pour faciliter la navigation. Pour le moment, le site qui a été réalisé est assez vide en terme de contenu. En effet, celui-ci sera davantage rempli au cours de la réalisation du projet. Ci-contre, nous pouvons retrouver la page *index* et la page d'*accueil*. La page *index* ne contient pas de texte. En effet, seuls le logo et une image de fond sont visibles. Quant à la page d'*accueil*, celle-ci présente très brièvement l'intérêt du site.



FIGURE 11 – Page index



FIGURE 12 – Page d'accueil

## 6 Réalisations de la première soutenance

Lors de la première soutenance, nous avions la possibilité d'ajouter des images à notre projet, ainsi que de les modifier avec différents filtres implémentés à l'aide de *pixbuf*. Le gestionnaire de calque nous permettait seulement d'ajouter et de supprimer des calques, cependant ceux-ci n'étaient pas encore reliés aux images que nous affichions à l'écran.

Nous pouvions déjà appliquer les fonctions de rotations d'image et d'effet miroir, ainsi que les fonctions de re-dimensionnement et de modification du curseur. Le créateur de filtre contenait seulement quelques boutons permettant d'avoir un aperçu de ses fonctionnalités futures, et pour finir, le site internet était terminé dans sa globalité.

## 7 Réalisation de la seconde soutenance

Pour cette deuxième soutenance, nous avons considérablement avancé dans notre projet. Nous avons pu ajouter de nombreuses fonctionnalités comme citées précédemment.

Un changement majeur de notre projet aura été l'amélioration du design de notre logiciel à l'aide d'un fichier *css* que nous avons créé. Ce fichier nous permet de modifier à notre convenance l'apparence des boutons, des fenêtres et des zones de saisies, ainsi que leur couleurs. Nous avons également ajouté différents boutons afin de rendre l'interface plus accessible à l'utilisateur.

Mis à part ce changement, nous avons ajouté différents filtres, afin que le logiciel permette de modifier le plus largement possible les images des calques que l'utilisateur importe. Nous avons également complété notre créateur de filtre afin que ce dernier permette de sauvegarder un filtre et de pouvoir en charger un depuis le menu de sélection de filtre.

Nous avons également fait en sorte que notre gestionnaire de calques soit relié à l'affichage de ces derniers. Cela nous permet donc d'afficher et de sauvegarder plus facilement ces calques. Il est même possible de restaurer une session sauvegardée au préalable et de reprendre la modification de l'image en cours. Le *GMPF\_LayerMngr* nous permet également de modifier l'ordre des calques, et de gérer les différents évènements liés aux mouvements de la souris, comme le dessin, la modification du curseur, etc.

Pour finir, nous avons pu ajouter la fonctionnalité de dessin sur le calque sélectionné. Ceci est également l'une des grosses modifications que nous ayons effectué sur notre projet. Cela permet à l'utilisateur de gommer ou bien de dessiner avec une brosse circulaire sur ses calques. Nous avons pour le moment implémenté deux types de brosses, circulaire et carré, et nous envisageons d'en ajouter d'autres pour la prochaine soutenance.

## 8 Objectifs pour la prochaine soutenance

Pour la prochaine soutenance, nous pensons implémenter les derniers outils présents dans la barre d'outils. Il serait alors possible d'interagir avec la portion de l'image sélectionnée, lui appliquer des filtres, dessiner uniquement dans cette zone, ou encore faire un copier-coller d'une partie de l'image dans un nouveau calque afin de le réutiliser ailleurs dans l'image.

Pour les filtres, tous les filtres pré-installés ayant déjà été réalisés, nous pensons ajouter la possibilité d'appliquer un filtre à tous les calques d'un seul coup, ainsi que la possibilité d'ajouter des filtres utilisant des matrices grâce à la fonction de convolution implémentée par nos soins.

## Conclusion

Nous sommes bien parti pour terminer notre projet et sommes en avance par rapport aux objectifs que nous nous étions fixés. La cohésion du groupe se fait bien même s'il est parfois difficile de s'organiser étant donné que nous sommes séparés dans deux classes différentes et que nos emplois du temps ne se concordent pas forcément.

Nous avons cependant réussi à réaliser chacun la partie qui lui avait été assignée dans le cahier des charges et n'avons pas rencontré de difficultés à communiquer et s'entraider pour que chacun acquière des connaissances durant la réalisation de ce projet. Nous avons donc bon espoir de rester soudés et de réussir à finir notre projet dans les délais indiqués.