

EPITA

Rapport de soutenance 3 GMPF

par *void**

Samory DIABY
Pierrick GUILLAUME
Arnaud CASSEZ
Sabine HU

22 mai 2019

Table des matières

Introduction	3
1 Retour rapide sur le cahier des charges	4
1.1 Planning	4
1.2 Répartition des tâches	5
2 Présentations	6
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
3 Fonctionnalités	10
3.1 Transformation de l'image	10
3.2 Interface	11
3.3 Gestion des calques	15
3.3.1 Gestionnaire des calques	15
3.3.2 Calques	16
3.4 Sauvegarde des calques	17
3.4.1 Exportation des calques	17
3.4.2 Sauvegarde des projets	17
3.5 Filtres	18
3.5.1 Historique	18
3.5.2 Explication illustrée	20
3.6 Effets	31
3.7 Créateur de filtres	32
3.8 Les actions "annuler" et "rétablir"	33
3.9 La sauvegarde automatique	34
3.10 Exemple d'images réalisées avec notre projet	36

4 Outils	37
4.1 Taille du curseur ajustable	37
4.2 Pinceau	37
4.3 Gomme	38
4.4 Sélection	38
4.5 Pipette	38
5 Site internet	40
5.1 Intérêt	40
5.2 Structure du site	41
6 Récapitulatif	44
7 Réalisations de la première soutenance	45
8 Réalisations de la seconde soutenance	47
9 Réalisations pour la dernière soutenance	49
Conclusion	52

Introduction

Nous nous étions fixés pour objectif de réaliser le projet *GMPF* sur une période de cinq mois. Nous voici arrivé à la dernière soutenance et nous avons chacun participé activement à la réalisation de notre idée, qui est désormais achevée avec succès. Nous avons pu réaliser tous les objectifs que nous nous étions fixés au départ et nous nous sommes même permis d'ajouter quelques fonctionnalités bonus qui améliorent grandement l'utilisation de notre application, mais nous donnerons tous les détails dans la partie réalisation du projet. Il s'agit lors de cette soutenance de présenter le travail qui a été réalisé, tout en citant les difficultés rencontrées.

Durant ces cinq mois, notre équipe a eu besoin de rester soudée afin de faire preuve de rigueur, mais aussi d'écoute envers chacun des membres afin de pouvoir s'aider mutuellement dans chaque problèmes rencontrés afin d'avancer de manière efficace dans la réalisation du projet.

1 Retour rapide sur le cahier des charges

1.1 Planning

	1 ^{ère} Soutenance	2 ^{ème} Soutenance	Soutenance Finale
Projet			
Calques			
Gestion de calques	50%	80%	100%
Rotation de calques/images	50%	100%	100%
Re-dimension de calques/images	50%	100%	100%
Filtres			
Créateur de filtres	50%	80%	100%
Filtres pré-installés	70%	100%	100%
Outils			
Taille du curseur ajustable	20%	50%	100%
Effets	20%	70%	100%
Sauvegarde			
Sauvegarde de calques	0%	50%	100%
Sauvegarde du projet (ensemble de calques)	0%	25%	100%
Interface graphique			
Interface	25%	60%	100%
BONUS			
Possibilité de dessiner		-	
Site internet			
Interface	33%	67%	100%
Téléchargement du projet	0%	50%	100%

1.2 Répartition des tâches

	Samory	Pierrick	Arnaud	Sabine
Projet				
Calques				
Gestion de calques	O		X	
Rotation de calques/images		O	X	
Re-dimension de calques/images		X		O
Filtres				
Créateur de filtres	X			O
Filtres pré-installés			O	X
Outils				
Taille du curseur ajustable	X	O		
Effets		O		X
Sauvegarde				
Sauvegarde de calques	X		O	
Sauvegarde du projet (ensemble de calques)	O		X	
Interface graphique				
Interface	O	X		
BONUS				
Possibilité de dessiner	-			
Site internet				
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 reprises 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 était de bien plus grande envergure et s'annonçait comme un défi encore plus grand pour moi.

La réalisation de ce projet m'a beaucoup plu. J'ai appris beaucoup de choses dans le langage *C* et sur la bibliothèque *Gtk* durant ce projet.

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. Désormais, le projet étant achevé, j'ai eu l'occasion d'acquérir de nouvelles connaissances durant ces cinq derniers mois.

2.2 Présentation du projet

Le groupe *void** est donc composé de :

Samory DIABY (samory.diaby) - notre chef de projet

Ainsi que de ses trois larbins :

- Sabine HU (sabine.hu)
- Pierrick GUILLAUME (pierrick.guillaume)
- Arnaud CASSEZ (arnaud.cassez)

Nous avons choisi de nommer notre projet *GMPF* pour *Graphic Maestro Photo Formaat*. Avec un nom pareil, impossible de se tromper, il s'agit bien évidemment d'un logiciel de création de musique!... ? Non

Autant pour moi. 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). Nous voulions avec un tel nom, montrer à quel point notre logiciel est polyvalent et qu'il correspond à tout le monde, quel qu'en soit son usage.

Description du projet

Notre projet est, comme dit précédemment, un logiciel de retouche et d'édition d'images. Il permet 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 avons fournis directement dans une interface graphique très intuitive et visuelle, afin qu'un utilisateur non expérimenté puisse s'y retrouver.

Le projet GMPF permet donc d'apporter des modifications aux images. Cependant, d'autres fonctionnalités sont également présentes. En effet, il est possible de sauvegarder un projet et de reprendre son édition plus tard. De plus, l'utilisateur peut exporter son projet une fois que ce dernier a été ter-

miné afin de combiner les différents calques et modifications en une seule image utilisable au format *PNG*.



3 Fonctionnalités

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 redimensionnement de l'image. La partie rotation a été effectuée à l'aide de la bibliothèque *Cairo*, initialement avec la bibliothèque *GTK* lors de la première soutenance. 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 avons utilisé 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 afin 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 re-dimensionner l'ensemble des calques afin de permettre à l'utilisateur d'agrandir l'image 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. Bien évidemment, le re-dimensionnement de l'image n'engendre désormais aucune perte de la qualité de l'image alors que lors de la première soutenance, une grande perte de qualité était observable dû à l'utilisation d'une fonction fournie par *GTK*.

3.2 Interface



FIGURE 1 – Interface

Ci-dessus, nous pouvons observer une image de l'interface finale du projet *GMPF*. Nous présenterons chaque élément présent, pour ce faire, la présentation se fera de haut en bas et de gauche à droite.

Tout d'abord, nous avons fait en sorte que lorsque l'utilisateur est en train de modifier une image ou même un projet, le titre de la fenêtre de notre application se met à jour afin d'afficher le nom du fichier utilisé, ainsi que les dimensions de celui-ci. Cela permet donc à l'utilisateur de se retrouver plus facilement. Nous retrouvons toutes ces informations tout en haut de la fenêtre, sur la même ligne que les boutons *réduire*, *agrandir* et *fermer*.

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

Nous avons également mis à disposition de l'utilisateur une série de boutons permettant d'effectuer certaines actions régulières sur l'image :

- Le premier bouton permet de choisir une couleur pour le pinceau, mais

aussi pour les filtres *Teinture* et *Accentuation des couleurs*.

- Le second bouton permet de choisir un angle de rotation pour le calque sélectionné. Une fois que l'utilisateur a choisi de peindre ou de gommer sur l'image, alors la rotation est définitive : l'image se met à jour et les parties de l'image qui se trouvent en dehors de la zone d'affichage se retrouvent rognées. Ce système permet à l'utilisateur de choisir un angle de rotation précis, mais aussi de continuer à dessiner normalement sur l'image une fois que la rotation a été effectuée. La rotation avait été correctement implémentée lors de la deuxième soutenance, seules des minimales modifications ont été apportées.
- Nous avons ajouté un troisième bouton servant à définir la taille du pinceau avec lequel l'utilisateur pourra dessiner sur l'image. Ce dernier était imposé lors de la deuxième soutenance.
- Le quatrième et le cinquième boutons sont liés. Ils servent respectivement à annuler et rétablir la dernière action. Cette fonctionnalité est extrêmement pratique et fondamentale, surtout lors de l'utilisation du pinceau puisqu'elle permet d'annuler un coup de crayon maladroit sans avoir à tout recommencer.
- Enfin, le dernier bouton permet de choisir le 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. Quelques problèmes avaient été rencontrés lors des précédentes soutenances mais aucun est observable depuis.

Sur la gauche, nous retrouvons, dans l'ordre suivant, les boutons permettant de sélectionner les outils : *Pinceau*, *Gomme*, *Pipette à couleur* et enfin

l'outil de *Sélection rectangulaire*. Ce dernier outil est relativement efficace et utile puisqu'il permet d'appliquer des filtres directement sur la partie sélectionnée et non à tout le calque sélectionné. L'utilisateur peut également effectuer les actions suivantes : *copier*, *coller*, *couper* et enfin *supprimer*.

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. C'est sur cette zone que l'utilisateur pourra peindre sur le calque qu'il aura préalablement sélectionné.

La partie droite de la fenêtre nous permet d'afficher les différents calques qui composent notre image. Nous avons donné à l'utilisateur la possibilité d'en retirer s'ils 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 comme il le souhaite grâce aux boutons situés sous cette liste. Les calques s'affichant les uns par dessus les autres, cela modifie l'image qui se trouvera au premier plan. Nous avons également ajouté une case cochable pour chaque calque qui définit si celui-ci doit apparaître ou non dans l'aperçu de l'image. Cette option peut être pratique si l'utilisateur souhaite se concentrer sur un calque en particulier sans être gêné par les autres. La case correspondant à l'affichage du calque ne sera donc pas cochée.

Sous cette liste, nous retrouvons un bouton permettant d'ajouter un calque sur notre image à partir d'une fenêtre qui apparaît lorsque nous appuyons dessus. Celle-ci permet de lui donner un nom, une taille ainsi qu'une position. Cette position sera appliquée comme un décalage du calque sur l'image. Il y a également un dernier bouton permettant de choisir une image sur laquelle

se superposera le calque. Si l'utilisateur choisit une image alors celle-ci sera rognée ou agrandie pour correspondre à la taille donnée. Dans le cas contraire, une surface vide sera créée afin de laisser l'âme de dessinateur de notre utilisateur s'exprimer.

Le deuxième et le troisième bouton permettent respectivement de déplacer d'un cran vers le haut ou le bas le calque sélectionné dans la liste. Le quatrième bouton permet de modifier les propriétés du calque par le biais d'une fenêtre similaire à celle pour ajouter un calque. Enfin, le dernier bouton permet de supprimer le calque sélectionné de la liste. Cependant, il est fort probable que l'utilisateur soit maladroit et supprime par erreur le mauvais calque. Ce dernier est chanceux puisqu'il est toujours possible de revenir en arrière à l'aide du bouton *Annuler* qui a été présenté précédemment.



FIGURE 2 – Ajout d'un calque

Afin d'être certain que chacun des utilisateurs soit satisfait de l'esthétique de notre interface, nous avons mis en place plusieurs thèmes. Ce dernier est à choisir dans la rubrique *Préférences*. Divers thèmes ont donc été mis en place afin de répondre aux exigences des utilisateurs. Parmi eux, nous retrouvons les deux thèmes suivants, le thème par défaut étant celui que nous visualisons dans les précédentes images.



FIGURE 3 – Thème bleu clair



FIGURE 4 – Thème vert sombre

3.3 Gestion des calques

Pour cette dernière soutenance, nous avons fini d'implémenter la gestion des calques, le fait de pouvoir modifier leurs propriétés, les déplacer sur l'image, ainsi que la gestion du retour arrière si l'on ajoute, supprime, modifie ou déplace un calque. Le gestionnaire des calques se situe directement sur l'interface, sur la partie droite de la fenêtre. Ainsi, il est à portée de l'utilisateur.

3.3.1 Gestionnaire des calques

Pour la première soutenance, le gestionnaire de calques, que nous avons décidé de nommer "GMPF_LayerMngr" pour "Manager de Calques de GMPF", 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 *GMPF*. Enfin, l'utilisateur a la possibilité de masquer un calque afin que celui-ci n'apparaisse plus sur l'aperçu.

3.3.2 Calques

Lors de la première soutenance, nous avons implémenté des *GMPF_Layer*. Ces derniers correspondent tout simplement aux calques. Nous avons décidé de les lier par une liste chaînée, avec en sentinelle le *GMPF_LayerMngr*. Pour sa part, le calque ne contient qu'une image accompagnée d'une position, d'une taille, d'un booléen 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. Grâce à l'interface que nous avons terminée, il est maintenant possible d'éditer les propriétés du calque après que celui-ci ait été créé. Nous pouvons également choisir de le masquer ou bien de lui appliquer un filtre.

Pour la première soutenance, nous utilisons la bibliothèque graphique *Gtk* pour gérer nos images. Nous nous sommes cependant rendu compte que cette bibliothèque nous ralentirait beaucoup sur la gestion des calques car il aurait fallu recréer une image à chaque fois que nous voulions mettre à jour notre aperçu. Nous avons donc 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 meilleure intégration des calques à notre image. En effet, l'image finale 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 *surface* présentes sur les calques, afin de les recréer sur une *surface* temporaire que nous transformons en *PNG* par la suite. L'image est ensuite sauvegardée avec le nom entré plus tôt par l'utilisateur.

Pour cette dernière soutenance, nous avons tenu nos promesses, notre gestionnaire de calques est extrêmement efficace et nous permet de gérer des images de grande taille (nous l'avons testé avec une image de 18000 * 18000 pixels). L'application de filtres à notre image est très rapide nous avons même ajouté plusieurs fonctionnalités bonus tels que le dessin et la gestion du retour en arrière, deux fonctionnalités très pratiques qui nous semblaient indispensables pour une application de retouche d'images.

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 ou du 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

3.5.1 Historique

Lors de la première soutenance, nous appliquions nos filtres directement à l'image en utilisant la bibliothèque *GTK*. En passant à la bibliothèque *Cairo*, il a fallu ré-implémenter tous les filtres déjà présents précédemment, mais aussi en ajouter de nouveaux pour compléter notre panoplie. Enfin, pour cette dernière soutenance, nous avons ajouté la possibilité d'appliquer le même filtre à chacun des calques d'un seul clic, mais aussi d'appliquer le calque uniquement à la partie sélectionnée pour donner plus de liberté à l'utilisateur et donc la possibilité de voir plusieurs filtres appliqués sur différentes parties d'une image. La majorité des filtres avait également été implémenté à l'aide de la bibliothèque *SDL* lors de la première soutenance car chaque partie du projet avait été réalisée séparément. L'utilisation de *SDL* a donc été principalement pour avoir un aperçu de l'application des filtres.

Lors de la première soutenance, les filtres que nous avons implémentés sont les suivants : *Teinture*, *Nuance de gris*, *Binarisation*, *Binarisation colorée*,

Négatif, Flou, Bords, Epine. Notons que les trois derniers filtres cités ci-dessus n'étaient pas encore reliés à l'interface. En effet, nous avons rencontré quelques problèmes qui sont désormais résolus.

Pour la seconde soutenance, nous avons implémenté plus de filtres comme *Chaud* et *Froid*. Nous avons également relié tous les précédent filtres à l'interface grâce à l'application de matrices de convolutions, ce qui permet d'appliquer plusieurs filtres différents en utilisant la même méthode. Nous avons alors prévu de donner la possibilité à l'utilisateur de créer ses propres matrices mais nous n'avons pas encore implémenté cette fonctionnalité.

Pour la dernière soutenance, nous avons permis de rendre l'application de filtres beaucoup plus rapide car l'application était plutôt lente sur les images de grande taille. Pour cela nous avons utilisé les *threads* qui nous permettent de diviser le travail entre les différents curs du processeur. Cette optimisation nous permet de réaliser l'effet "bordure" en moins de 6 secondes sur une image de 16000 x 4700 pixels.

3.5.2 Explication illustrée

Dans cette partie, nous allons présenter les filtres qui ont été implémentés depuis le commencement du projet. 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 5 – Image d'origine

Niveau de gris

Le filtre *Niveau de gris* permet de changer une image en couleur en une image composé de niveau de gris. Pour ce filtre, nous effectuons la moyenne des valeurs RGB de chaque pixel afin d'obtenir la valeur de gris du pixel en question.

Binarisation

Le filtre *Binarisation* permet de modifier une image en couleur afin que cette dernière soit composée uniquement de pixels noirs ou blancs. Pour ce filtre, nous avons utilisé un seuil de 127, qui représente la moitié de la valeur que peut prendre un pixel. Ainsi, nous regardons la valeur de chaque pixel

FIGURE 6 – Filtre *Niveau de gris*

puis nous la comparons au seuil afin de déterminer si ce pixel sera blanc ou noir.

FIGURE 7 – Filtre *Binarisation*

Binarisation couleur

Le filtre *Binarisation couleur* est semblable au filtre *binarize*. En effet, celui-ci prend en compte chacune des valeurs RGB du pixel, enfin si l'une des valeurs est supérieure à 127, la valeur R, G ou B prendra la valeur 255. Dans le cas contraire, la valeur sera de 0. On peut ainsi constater que l'image est colorée.

FIGURE 8 – Filtre *Binarisation couleur*

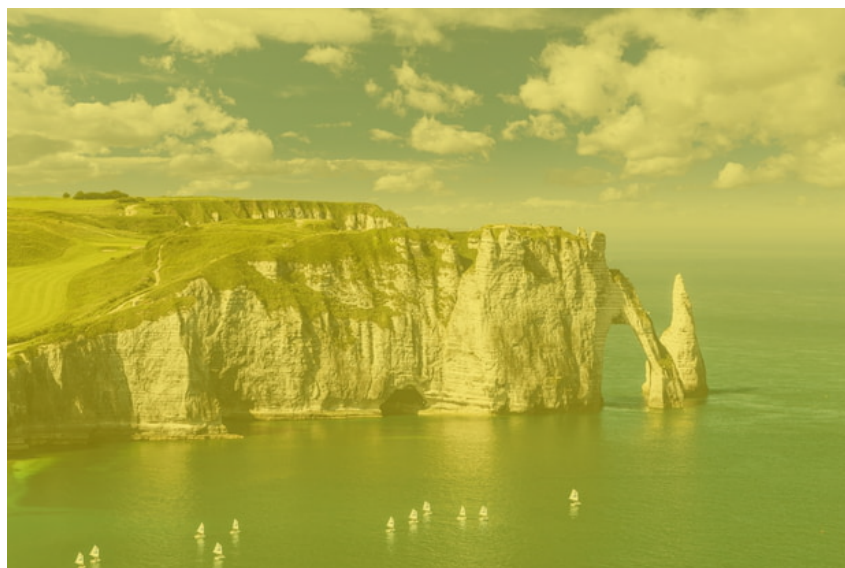
Negative

Le filtre *Negative* permet d'inverser la couleur de chaque pixel de l'image. L'implémentation de ce filtre n'a donc posé aucun problème puisqu'il suffit de associer à l'image les valeurs RGB résultantes de la différence des valeurs maximales et des valeurs RGB du pixel.

FIGURE 9 – Filtre *Negative*

Tinter

Le filtre *Tinter* permet à l'utilisateur de choisir une couleur et de l'appliquer avec un certain pourcentage à l'image présente sur le calque actuel. Pour ce faire, l'utilisateur pourra activer le bouton lié au filtre, et se verra proposer de choisir une couleur par le biais d'une boîte de dialogue apparaissant à l'écran.

FIGURE 10 – Filtre *Tinter* en jaune

Flou

Le filtre *Flou*, correspond à un effet de flou. Ce filtre applique une matrice de flou, stockée sous la forme d'une structure, à l'ensemble des pixels de l'image. Nous sommes donc obligés de créer une nouvelle image où l'on stocke les pixels résultant de la transformation puis remplaçons l'image de départ par celle-ci. Ce filtre à beaucoup gagné en performance grâce à l'usage du multicurs.

FIGURE 11 – Filtre *Flou*

Epine

Le filtre *Epine*, comme le filtre *Flou*, s'effectue de la même manière. En effet, seule la matrice donnée est différente, la précédente permettait d'obtenir une image flou et celle-ci nous permet d'obtenir une image dont les contours sont bien plus accentués.



FIGURE 12 – Filtre *Epine*

Bords

Le filtre *Bords* est semblable aux deux derniers. En effet, seule la matrice est différente. Ce filtre-là nous permet d'obtenir une image très assombrie dont seuls les contours sont de couleurs claires.

FIGURE 13 – Filtre *Bords*

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 14 – 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 15 – Filtre *Froid*

Vert

Le filtre *vert* s'inspire des deux filtres précédents. En effet, l'un d'eux se concentrait 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 16 – Filtre *Vert*

Luminosité +

Le filtre *luminosité +* fait apparaître l'image plus claire. 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 17 – 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 18 – 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.

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

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

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

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 19 – Filtre *Égalisation en niveau de gris*

Égalisation en couleurs

Le filtre *égalité 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. Nous avons donc besoin d'allouer trois fois plus de mémoire afin d'appliquer ce filtre. A l'issue de l'application du filtre, nous obtenons l'image suivante.

FIGURE 20 – Filtre *Egalisation en couleurs*

3.6 Effets

Comme nous avons pu le voir précédemment, la rotation ainsi que les miroirs ont été mis en place. Ci-dessous, nous retrouverons les deux effets miroir implémentés, selon un axe vertical ou horizontal. Leur implémentation a été assez simple. En effet, il suffit d'échanger les valeurs de deux pixels.

FIGURE 21 – Effet *Miroir Horizontal*

FIGURE 22 – Effet *Miroir Vertical*

3.7 Créateur de filtres

Pour la deuxième soutenance, nous avons poursuivi l'implémentation du créateur de filtre afin que celui puisse correspondre au mieux à ce que nous avons indiqués dans le cahier des charges. Nous avons donc pour cela ajouter la possibilité d'ajouter un deuxième filtre afin d'augmenter le nombre de combinaisons possible.

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

Désormais, le créateur de filtres est différent. En effet, il s'appuie sur le même principe que les filtres de convolution. Une matrice de taille 3x3 ou 5x5 est présente dans la fenêtre. L'utilisateur entre lui même les valeurs. De plus, il peut tester sa matrice de convolution sur l'image avant de l'appliquer. Un aperçu de l'image est disponible.

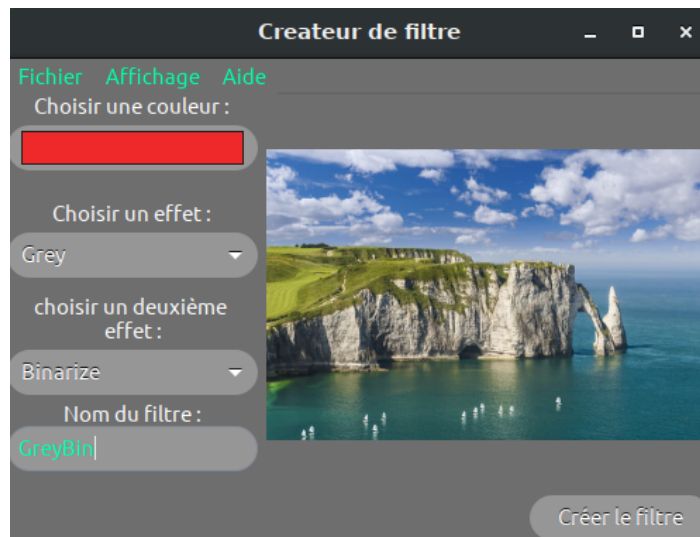


FIGURE 23 – Créateur de filtre lors de la deuxième soutenance

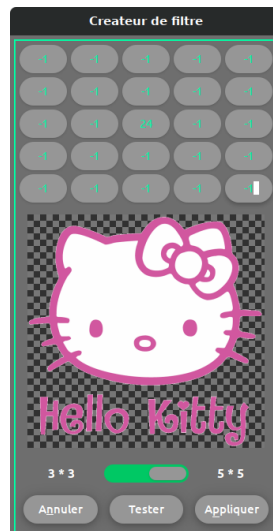


FIGURE 24 – Créateur de filtre actuel

3.8 Les actions "annuler" et "rétablir"

Pour cette dernière soutenance, nous avons également implémenté un système permettant d'annuler ou de rétablir les différentes actions de l'utilisateur. Celui-ci marche avec un tableau d'actions ainsi que des fichiers temporaires créés à l'aide de la fonction `tmpfile()` de la bibliothèque standard du C. Le tableau contient les *files descriptor* des fichiers temporaires. Ces derniers contiennent les données relatives aux précédentes actions. Ceux-ci nous permettent de sauvegarder l'état du projet en ne gardant que les informations essentielles.

Ainsi nous sauvegardons le calque en entier uniquement lorsque nous modifions l'image, uniquement ses propriétés (nom, taille et position) lors de leur modification à l'aide de la fenêtre prévu à cet effet, et uniquement l'action avec son calque associé pour une modification de l'ordre des calques (une place plus haut/bas dans la liste). Cet ajout a été très important mais nous a semblé réellement indispensable dans une application de ce genre où une mauvaise manipulation sans cette fonctionnalité peut vous amener à perdre votre travail depuis votre dernière sauvegarde.

3.9 La sauvegarde automatique

Enfin, nous avons pensé à ajouter une dernière fonctionnalité, relativement simple mais qui peut sauver des heures de travail : la sauvegarde automatique. Le principe est simple, à chaque action de l'utilisateur, l'application va écraser l'ancienne version de la sauvegarde automatique, permettant alors de conserver en permanence la dernière version possible, contenant toutes les modifications non sauvegardées. Cependant, nous ne nous sommes pas arrêtés à cette étape. En effet, lorsque l'utilisateur sauvegarde son projet, cela signifie que toutes les modifications qu'il souhaitait apportées ont été sauvegardées. Il n'y a donc plus de raison de garder une copie des mêmes informations. Nous détruisons alors le fichier. La même chose est faite lorsque l'utilisateur quitte sans sauvegarder les modifications. Cela signifie qu'il ne souhaite pas les conserver, nous pouvons donc les supprimer. Ainsi lorsque l'utilisateur quitte l'application de manière conventionnelle (sans que le programme ou l'ordinateur s'arrête, alors le fichier de sauvegarde n'existe plus. Ce n'est que lorsque l'application s'est fermé sans intervention de l'utilisateur que le fichier reste. Grâce à cela, lorsque l'utilisateur ouvre un projet, l'application vérifie qu'il n'existe pas de fichier de sauvegarde automatique. Si un tel fichier n'existe pas, alors elle charge simplement le projet. Sinon, l'application ouvre une fenêtre demandant si l'utilisateur souhaite charger la sauvegarde automatique et la charge en cas de réponse positive. Cette fonctionnalité a pris un certain

temps à implémenter car il a fallu repenser la façon dont l'ouverture d'un projet était gérée mais celle-ci reste très pratique, surtout pour les utilisateurs qui ne pensent pas toujours à sauvegarder après chaque modification importante.

3.10 Exemple d'images réalisées avec notre projet

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 25 – Images retouchées - Soutenance 2



FIGURE 26 – Image retouchée - Soutenance 3

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. Nous avons également ajouté un outil de sélection.

4.1 Taille du curseur ajustable

Pour cette dernière soutenance, nous nous sommes concentrés sur l'amélioration du système de *curseur ajustable*. Nous avons donc désormais la possibilité de changer la taille curseur lorsque nous nous déplaçons sur la zone de modification de calque, et ce même lorsque ce dernier est redimensionné. Nous avons également ajouté différents curseurs de forme carré, rectangulaire et triangulaire, qui vont servir à l'utilisation des nouveaux pinceaux que nous décrirons plus bas.

Ces curseurs permettent donc toujours de dessiner à l'aide des pinceaux 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 dernière soutenance, nous avons pu implémenter de nouveaux pinceaux, qui sont les pinceaux triangulaire, rectangulaire et carré.

L'outil pinceau permet à présent à l'utilisateur de dessiner sur le calque qu'il a sélectionné de différentes manières. Il pourra toujours choisir une couleur avec le bouton de sélection de couleur *GTK* et choisir une taille de pinceau à l'aide du bouton de redimensionnement du pinceau.

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. Pour cette soutenance, nous avons seulement ajouté de nouvelles formes de curseur à cet outil.

4.4 Sélection

Pour cette soutenance, nous avons ajouté l'outil *sélection* qui comme son nom l'indique, permet de sélectionner une partie de l'image afin d'effectuer différentes actions. Nous avons donc la possibilité de copier le morceau d'image sélectionné afin de le coller sur une autre partie du calque, voire sur un autre calque. Nous avons également la possibilité d'appliquer des filtres sur ce morceau d'image sans modifier le reste de l'image.

4.5 Pipette

L'outil pipette a également été ajouté pour cette soutenance, et permet donc de sélectionner la couleur d'un pixel présent sur le calque. La couleur sélectionnée est stockée dans le bouton de sélection de couleur situé en haut à gauche de l'interface.

A partir de cette outil, nous avons implémenté deux fonctions qui sont *color_killer* et *swap_color*, qui permettent respectivement de supprimer un ensemble de couleur et de remplacer un ensemble de couleur. Ces deux fonctions peuvent être appelées par le biais du menu outil grâce aux boutons *Supprimer une couleur* et *Remplacer une couleur*. Bien que nous avons conscience que ses deux fonctions, notamment la suppression d'une couleur, sont peu utiles. Il est toujours préférable de les avoir à disposition. En effet, pour la suppression

d'une couleur, les pixels sont remplacés par un pixel transparent. De plus, ces deux fonctions modifient seulement les pixels possédant exactement les mêmes valeurs rouge, vert et bleu.

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. Tout se trouve sur une seule page et à portée de clic. Nous avons choisi un style très épuré qui correspond à l'interface de notre application : une interface simple d'utilisation mais riche en fonctionnalités. De plus, nous avons repris le même code couleur que le thème principale de notre application à savoir bleu sarcelle.



FIGURE 27 – Site internet

Lorsque l'on arrive sur notre site, nous trouvons tout d'abord une grande image avec notre logo. Juste au dessus se trouve la bar de navigation. Celle-ci nous permet d'accéder rapidement à n'importe quelle section de notre site en un clic. On peut également distinguer un bouton en bas à gauche de l'image. Si l'on clique dessus, cela affiche une fenêtre modale affichant une brève description de notre groupe ainsi que de notre groupe.

Dans la continuité, nous trouvons une ligne sur laquelle se trouve une photo de chaque membre du groupe. Lorsque l'on clique dessus, cela ouvre la description de la personne concernée.

Juste en dessous nous retrouvons la description complète de notre projet ainsi que l'intérêt de celui-ci pour nous.

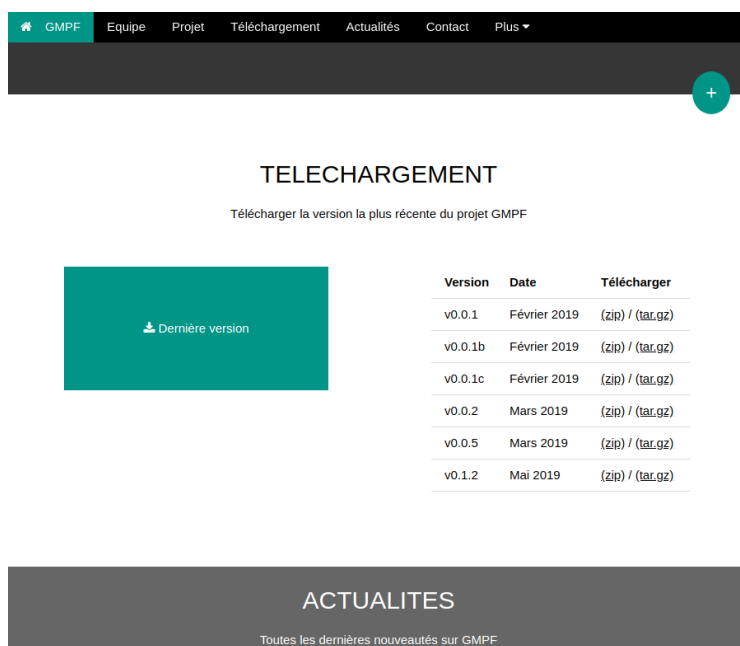
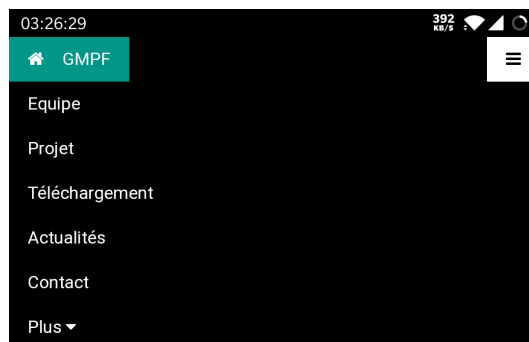


FIGURE 28 – Téléchargement

Nous voilà arrivé à la partie *téléchargement*. Nous pourrions bien évidemment y télécharger le logiciel dans sa dernière version mais également le code source des anciennes versions que les utilisateurs pourrions compiler pour obtenir toutes les versions précédentes. Cela inclut donc les versions de nos précédentes soutenances.

5.2 Structure du site

La structure du site est restée très simple. En effet, le but d'un site internet est d'aider les utilisateur et de leur permettre une utilisation plus simple de l'application. Il ne s'agirait pas de les perdre dans un dédale de pages sans queue ni tête. Le site contient toutes les informations importantes, ainsi que la possibilité de visionner directement les fichiers des soutenances précédentes sur le site grâce au lecteur de PDF intégré. Notre site internet a également été pensé pour les téléphone et s'adapte parfaitement à une utilisation sur mobile.



Samory Diaby

Chef de projet



Pierrick Guillaume

Larbin

FIGURE 29 – Interface sur téléphone

Le site internet a énormément évolué depuis la deuxième soutenance. En effet, la précédente version du site comportait plusieurs pages, chacune correspondant à une rubrique. Bien que ces deux versions sont très simple d'utilisation, celle que nous possédons désormais apparaît comme étant plus moderne et probablement plus agréable à naviguer.



FIGURE 30 – Page *index* de la précédente version du site.



FIGURE 31 – Page d'accueil de la précédente version du site.

6 Récapitulatif

Au terme des cinq mois, le projet est mené à bien. Celui possède de nombreuses fonctionnalités et permet réellement de modifier ses images.

Ci-dessous, nous retrouvons toutes les fonctionnalités du projet :

- Interface simple d'utilisation et complète.
- Chargement et sauvegarde d'une image, d'un calque. (Sauvegarde automatique).
- Possibilité de dessiner, divers pinceaux sont à disposition.
- Modification de l'image ou de l'ensemble du projet à l'aide de filtres (16 filtres).
- Gestionnaire de calques (ajout, modification de l'ordre, aperçu, taille ajustable, décalage).
- Sélection d'une partie de l'image.
- Rotation de l'image et miroirs.
- Ajout de thèmes de l'interface.
- Fonctionnalités *Copier*, *Coller*, *Couper* et *Annuler*, *Rétablir*.
- Agrandissement.
- Sélection et suppression de couleurs.
- Créateur de filtres.
- Divers curseurs.

7 Réalisations de la première soutenance

Lors de la première soutenance, l'objectif principal était d'avoir la structure globale du projet. Il nous était donc fondamentale d'avoir une interface graphique sur laquelle il est possible d'ouvrir une image et d'avoir un aperçu de cette dernière. De plus, il fallait également commencer l'implémentation des filtres ainsi que la gestion des calques. L'implémentation des filtres s'est effectuée à l'aide de *Pixbuf*, de la bibliothèque *GTK*. Le début de l'implémentation des filtres s'étaient effectuée sous *SDL* car cette bibliothèque était plus appropriée pour visualiser directement les images. Il a donc fallu transcrire ce qui avait été réalisé. Quant au gestionnaire de calques, ce dernier nous permettait seulement d'ajouter et de supprimer des calques. Cependant, ils n'étaient pas encore reliés aux images que nous affichions à l'écran.

D'autres fonctionnalités avaient également été ajoutées telles que la rotation d'image, les effets miroirs, le re-dimensionnement de l'image et la modification du curseur. Concernant la rotation d'image, celle-ci effectuait seulement des rotations à angle droit, et pour les effets miroirs, ils retournaient l'image selon un axe vertical ou horizontal. Le re-dimensionnement de l'image correspond principalement à un grossissement de l'image depuis l'interface. Cela permet ainsi à l'utilisateur d'observer les pixels de plus près et donc, d'effectuer un travail bien plus précis.

Concernant le créateur de filtres, il y avait une fenêtre qui apparaissait après avoir appuyé sur le bouton concernant le créateur. L'utilisateur avait la possibilité de choisir les caractéristiques de son propre filtre afin de les appliquer à l'image.

Quant au site internet, celui-ci était très simpliste. En effet, il possédait une interface simple qui permettait d'accéder aux différentes informations facilement. Bien évidemment, il n'était pas encore complet et certaines pages

n'étaient pas encore créées. Bien que celui-ci était bien structuré, nous avons choisi pour la dernière soutenance de changer entièrement son style. Ce dernier est actuellement bien plus épuré et plus simple d'accès, son aspect est plus moderne.

8 Réalisations de la seconde soutenance

Pour la 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.

Hormis la modification de l'interface, d'autres changements se sont également opérés. En effet, des nouveaux filtres sont été ajoutés à l'interface permettant à l'utilisateur d'avoir un plus large choix parmi les filtres présents. Concernant le créateur de filtre, celui-ci a également été complété. En effet, ce dernier pouvait désormais sauvegarder un filtre et 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 des images. 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. Le gestionnaire de calques a donc été fortement avancé pour cette deuxième soutenance.

Pour finir, nous avons pu ajouter la fonctionnalité de dessiner 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.

Quant au site internet, celui-ci a reçu de légères modifications. En effet, certaines pages ont été créées et d'autres ont été complétées. Dans la rubrique *téléchargement*, il était possible de télécharger la première version du projet, présentée lors de la première soutenance, ainsi que la deuxième version du projet, présentée lors de la deuxième soutenance.

9 Réalisations pour la dernière soutenance

Nous voilà déjà arrivé à la troisième soutenance et nous avons réalisé l'intégralité de nos objectifs fixés et nous avons même ajouté plusieurs fonctionnalités bonus afin de rendre notre application beaucoup plus agréable à utiliser, notamment grâce à notre système d'action de retour en arrière ainsi que de retour en avant. Nous avons également ajouté la possibilité de dessiner sur notre image car une application permettant seulement de modifier une image à l'aide de plusieurs filtres peut rapidement lasser les utilisateurs. Le dessin est donc une partie primordiale du projet, celui-ci permet à l'utilisateur d'effectuer bien plus de modifications et d'être plus libre lors de la retouche de l'image.

Pendant ce dernier mois, nous avons ajouté plusieurs outils très pratiques : nous avons implémenté un outil de sélection de couleur. Celui-ci permet à l'utilisateur de cliquer sur l'image et d'obtenir la couleur sur laquelle il a cliqué comme couleur de pinceau. Cet outil est très pratique puisqu'il n'a plus besoin d'aller chercher à reproduire une couleur en trifouillant dans le menu de sélection de couleur pendant plusieurs minutes sans forcément y parvenir.

Le second outil que nous avons implémenté est celui de la suppression de la couleur sélectionnée. Cet outil permet donc de choisir une couleur, souvent à l'aide de la pipette, puis de supprimer tous les pixels de la même couleur que celui sélectionné. Cela évite de devoir effectuer un travail fastidieux à la gomme lorsqu'une couleur ne convient pas à l'utilisateur par exemple.

Le dernier outil que notre équipe a ajouté à notre application est l'outil de sélection. Il permet de sélectionner une zone sur le calque sélectionné dans la liste délimitée par un rectangle rouge. Il permet donc d'interagir uniquement avec cette partie de l'image, de lui appliquer des filtres et d'effectuer les actions de copier, coller la sélection à un autre endroit ou sur un autre

calque, de couper la sélection ou bien simplement de la supprimer.

Nous avons également ajouté la possibilité de masquer un calque depuis l'interface. Le calque masqué n'apparaît plus dans l'aperçu de l'image afin de laisser l'utilisateur se concentrer uniquement sur la partie de l'image qu'il est en train d'éditer. Les calques ont également des noms, ce qui permet de les différencier et donc permet à l'utilisateur de s'y retrouver plus facilement, surtout que nous avons ajouté une barre de recherche qui permet de n'afficher dans la liste que les calques dont le nom contient la recherche. Cette possibilité de nommer ses calques est très utile dans les projets volumineux où le nombre de calques peut parfois devenir réellement important. Nous avons ajouté toutes ses fonctionnalités directement dans l'interface et elles sont disponibles d'un simple clique pour rendre notre application la plus simple d'utilisation possible. Nous n'avons cependant pas perdu en fonctionnalités pour autant.

Le gestionnaire des calques est donc terminé et offre de nombreuses fonctionnalités pratiques pour l'utilisateur afin de lui fournir une interface convenable pour la retouche d'une simple image comme pour le travail sur un projet beaucoup plus conséquent.

Nous avons également ajouté un système de annuler/rétablir grâce à différents types d'actions. Notre système permet d'enregistrer chaque action dans un fichier temporaire prévu à cet effet. Ce système permet donc d'enregistrer les cinq cents dernières actions de l'utilisateur et de les restaurer dans l'ordre. Même si cette fonctionnalité ne faisait pas partie de nos objectifs de départ, nous voulions obtenir une application d'édition d'images qui serait facile et pratique à utiliser. Or sans système de cette sorte, il faudrait recommencer l'édition depuis zéro à chaque petite erreur comme un coup de crayon maladroit ou simplement une suppression malencontreuse d'un calque nécessaire au projet. Il nous a fallu du temps pour implémenter ce système mais nous avons réussi avec brio. Nous avons choisi d'utiliser les fonctions déjà présent

dans la bibliothèque standard du C pour avoir un maximum de contrôle sur ce que nous faisons. Nous avons créé différents types d'action de l'utilisateur, et les ajoutons à notre liste de fichiers temporaires dès que nous détectons une entrée utilisateur. Ainsi, lorsque l'utilisateur souhaite revenir sur ses pas, nous rechargeons uniquement les informations stockées dans notre fichier. Cela permet par exemple si l'utilisateur ne souhaite que changer le nom d'un calque, de ne pas sauvegarder tout le calque mais uniquement l'ancien nom pour pouvoir le restaurer. Ce système limite donc l'utilisation de l'espace mémoire pour ne pas saturer notre disque dur.

Pour la première fois dans notre application, il est possible d'ajouter des calques avec une taille et une position définie. Ceci permet alors d'ajouter des calques à une position souhaitée. Par exemple, si l'utilisateur souhaite ajouter une image comme calque mais dans le coin inférieur droit de l'image, ceci est désormais possible grâce à cette nouvelle fonctionnalité. Si l'utilisateur souhaite modifier les propriétés du calque ultérieurement, cela pourra également se faire.

Pour ajouter un peu de personnalisation, nous avons donné la possibilité à l'utilisateur de changer de thème. Ceci se fait dans le menu Préférence > Thème. Nous offrons alors à l'utilisateur de créer son propre thème pour convenir au mieux à ses préférences.

L'ajout de toutes ces fonctionnalités nous a pris beaucoup de temps mais nous sommes désormais fiers de notre travail et de pouvoir vous le présenter aujourd'hui terminé.

Conclusion

Nous avons terminé notre projet en temps et en heure, avec même l'ajout de plusieurs fonctionnalités qui n'étaient pas prévues dans le cahier des charges. Nous pensons néanmoins que ces dernières sont essentielles pour obtenir une applications pratique et facile d'utilisation. Nous avons travaillé de manière coordonnées afin de s'entraider et obtenir le fruit de notre collaboration que nous vous présentons aujourd'hui. Nous avons eu une bonne cohésion de groupe même s'il a parfois été difficile de trouver des moments pour travailler ensemble étant donné que nous ne sommes pas tous dans la même classe.

En terminant ce projet, nous avons bon espoir de continuer notre cursus à EPITA en première année d'ingénieur l'année prochaine. Ce projet nous aura permis d'avoir une idée plus concrète du travail d'équipe sur un projet de longue durée, en effet, cela est complètement différent d'un TP et l'aide du camarade se distingue davantage. Nous espérons donc conclure notre cursus préparatoire par ce projet qui nous aura appris à chacun l'avantage d'avoir une équipe de travail efficace. Nous espérons également pouvoir continuer à travailler ensemble sur d'autres projets dans le futur.

Merci beaucoup.

Samory DIABY, Sabine HU, Arnaud CASSEZ et Pierrick GUILLAUME, le groupe Void *