

SOUTENANCE 2

2021-2022



Projet S4



On est là alors

Alexandre Devaux-Riviere

Kaël Facon

Grégoire Vest

Mathis Rabouille

Table des matières

1 Rappels - Projet Decolor	3
1.1 Notre Groupe	3
1.2 Le sujet du projet	3
1.3 Le choix des bibliothèques	3
1.3.1 Simple DirectMedia Layer (SDL) et (SDLgfx)	3
1.3.2 GIMP Toolkit - GTK	3
2 Avancements - Projet Decolor	4
2.1 Interface et ergonomie	4
2.1.1 Raccourcis claviers	4
2.1.2 Fonctionnalités Abandonnées	4
2.1.3 Nouvelles fonctionnalités	4
2.1.4 Apparence et thème CSS	6
2.2 Nouveaux outils et modification des anciens	11
2.2.1 Le crayon	11
2.2.2 Les outils de transformations de l'image	11
2.2.3 Les filtres	16
2.2.4 La pré-visualisation des formes et des outils	18
2.2.5 Fonctionnalité abandonnée	19
3 Site internet du logiciel	20
3.1 La structure du site	20
3.2 Page d'accueil	20
3.3 Page d'avancements	21
3.4 Page de documentation	21
3.5 Page de l'équipe	22
3.6 Page de téléchargement	23
4 Découpage du projet	24
4.1 Tableau de répartition des tâches	24
4.2 Prévisions soutenances	24
5 Développement sur les tâches réparties et le travail	25
5.1 Schémas Github - Contributions de chacun	25
5.2 alexandre.devaux-riviere	25
5.3 kael.facon	25
5.4 gregoire2.vest	26
5.5 mathis.rabouille	26
6 Nous contacter	26

1 Rappels - Projet Decolor

1.1 Notre Groupe

- Kaël Facon
- Grégoire Vest
- Mathis Rabouille
- Alexandre Devaux-Riviere (Chef de groupe)

Notre groupe est composé de 4 ex-lettions ayant dû rentrer en France suite au début du conflit entre la Russie et l'Ukraine. Nous nous sommes rassemblés au sein du groupe "On est là alors" car nous avions des idées en commun, nous voulions tous les quatre faire un logiciel en rapport avec le dessin et l'édition d'images. C'est donc dans cette optique que nous avons pensé à Decolor un logiciel linux.

1.2 Le sujet du projet

L'application (donc nommée Decolor) sera principalement orientée vers un logiciel très simple d'utilisation permettant à l'utilisateur de dessiner ou modifier une image afin de laisser libre cours à sa créativité. L'interface graphique de l'application, pour la partie outils, a une fonction qui permet de charger une image à partir d'un explorateur de fichier et de faire des modifications sur cette dernière ou de la manipuler. Par la suite, cette image peut être sauvegardée dans les fichiers locaux de l'ordinateur ou écrasée (au choix). Nous avons opté pour une interface sobre, ergonomique et surtout facile d'utilisation. L'intérêt algorithmique de notre projet se tourne vers l'imagerie, notamment vers le traitement et l'édition d'images comprenant la modification et le stockage de pixels.

1.3 Le choix des bibliothèques

1.3.1 Simple DirectMedia Layer (SDL) et (SDLgfx)

SDL :

Simple DirectMedia Layer est une bibliothèque de développement multiplateforme libre conçue pour fournir un accès de bas niveau au matériel graphique. Elle nous permet de travailler sur la manipulation d'images. Nous avons déjà pu utiliser cette bibliothèque lors de différents TPs ou encore lors de notre projet de troisième semestre qui consistait à résoudre un sudoku en faisant un OCR.

SDLgfx :

Le code de la bibliothèque SDLgfx fournit des routines de dessin de base et ajoute des fonctions utiles pour le zoom images et fait des traitements d'image de base sur des tableaux d'octets.

1.3.2 GIMP Toolkit - GTK

GTK est une boîte à outils multiplateforme gratuite et libre pour créer des interfaces utilisateur graphiques. Elle nous permettra de pouvoir réaliser une application simple et facile d'utilisation. Nous avons jugé que GTK serait l'outil parfait pour mener à bien nos ambitions.

2 Avancements - Projet Decolor

2.1 Interface et ergonomie

2.1.1 Raccourcis claviers

Dans le but de peaufiner l'ergonomie du logiciel, nous avons réfléchi à ce que nous pourrions ajouter. Nous avons donc décidé qu'à l'image des logiciels classiques de ce genre nous allions implémenter des raccourcis clavier pour pouvoir annuler et de nouveau appliquer des modification faite plus rapidement. Pour ce faire nous avons utilisé GTK qui possède une gestion du clavier intégrée, il existe un événement qui, lorsqu'il est déclenché, permet de récupérer l'identifiant de la touche pressée. Nous utilisons également une autre fonctionnalité de GTK : les masques qui nous permettent entre autre de gérer la souris mais également certaines touches du clavier spécifiques comme shift ou ctrl. Cela nous permet donc de détecter l'appui de n'importe quelle touche et de savoir si l'une des touches spéciales est appuyée en même temps, si c'est le cas alors on appelle la fonction associée au raccourci.

Cette fonctionnalité n'était pas parmi les plus urgentes à implémenter mais le logiciel étant dans un stade assez satisfaisant et le temps restant étant plutôt court nous avons décidé de commencer par ça au niveau de l'ergonomie. Désormais l'interface est assez complète et nous ne pouvons pas nous permettre de la surcharger de boutons et de fonctionnalités, pour cette raison les raccourcis clavier peuvent en effet s'avérer primordiaux. Tout les logiciels de dessins basent une énorme partie de leur ergonomie d'utilisation sur les raccourcis clavier

2.1.2 Fonctionnalités Abandonnées

Nous avions en tête d'ajouter la gestion de l'opacité au projet malheureusement à cause de certains choix que nous avions fait à la soutenance précédente. Cela était trop compliqué pour être faisable sans tout recommencer et nous n'avions pas du tout le temps de l'implémenter entièrement. En effet, plus tôt dans le projet nous avions choisi de garder SDL pour nos modifications d'image et que nous ne sommes pas passés à cairo pour interagir directement avec la zone de dessin. Notre version de SDL ne nous permettant pas de gérer la composante alpha sur nos couleurs nous a totalement empêché de développer cette fonctionnalité.

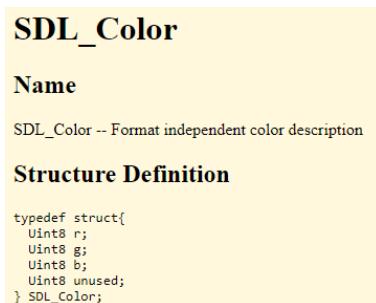


FIGURE 1 – Structure SDL Color dans notre version

2.1.3 Nouvelles fonctionnalités

Avec l'arrivée de nombreuses nouvelles fonctionnalités concernant le déplacement de l'image, ça rotation et aussi le redimensionnement de la zone de dessin il a fallu réfléchir à comment nous pouvions intégrer toutes ces fonctionnalités sans surcharger l'interface et la rendre illisible ou difficile à utiliser. Nous avons

donc après mures réflexions fini avec une interface assez chargée mais tout a fait lisible et utilisable simplement.

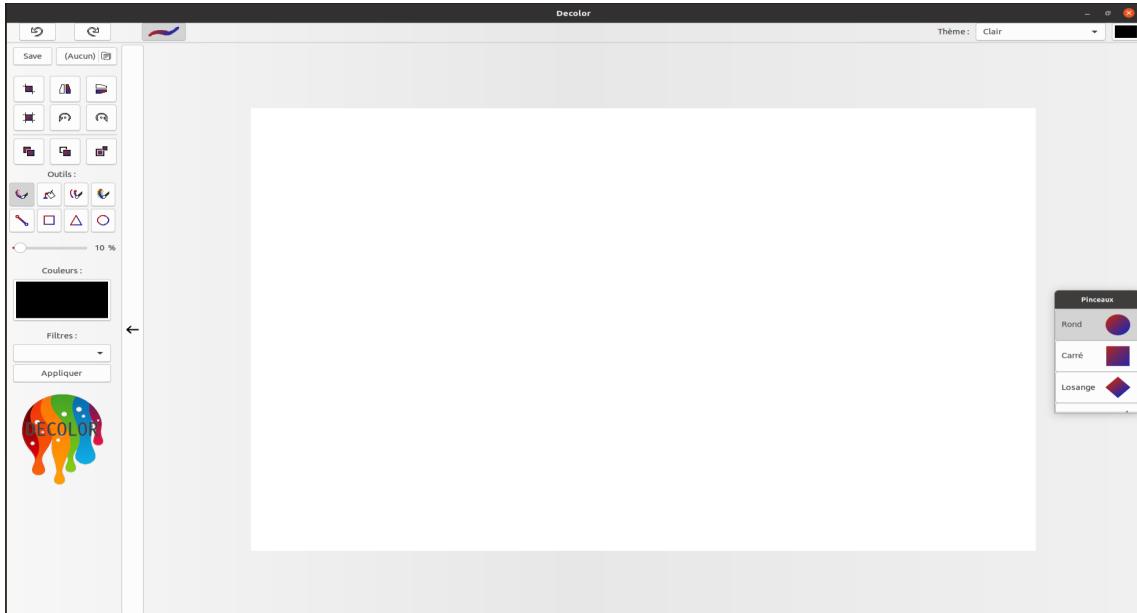


FIGURE 2 – Interface finale

Nous avons commencé par placer les nouveaux boutons dans l'interface pour accueillir les nouvelles fonctions. Nous les avons placés arbitrairement au dessus des huit premiers outils.

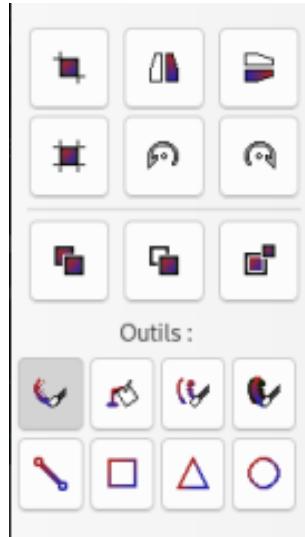


FIGURE 3 – Nouveaux boutons

Pour certaines des fonctionnalités il a également fallu implémenter de nouvelles fenêtre surgissante (pop-up). Dans le cas de la fonction de redimensionnement nous avions besoin d'une fenêtre pour récupérer les nouvelles dimensions de l'image auprès de l'utilisateur de façon efficace. Pour créer cette fenêtre pop-up nous avons utilisé un GTK Dialog, il nous permet de lancer une fenêtre personnalisable avec deux zones utilisables, une zone de contenu où l'on peut mettre n'importe quel type d'élément cela peut aller des

boutons aux entrées de texte ou simplement à du texte pour afficher un message et une zone d'action qui ne peut contenir que des boutons majoritairement utilisés pour valider un choix ou fermer la fenêtre en annulant l'action. La fenêtre basique créée pour la fonction de redimensionnement a, plus tard, été largement améliorée et de nouvelles fonctionnalités lui ont été ajoutées, comme le choix du placement de l'ancienne image dans la nouvelle zone agrandie.



FIGURE 4 – Fenêtre surgissante de redimensionnement

Pour rendre l'utilisation plus facile nous avons également corrigé quelques problèmes dont le fait qu'avant il fallait forcément rajouter l'extension .bmp au nom du fichier que nous étions entrain de sauvegarder, aujourd'hui l'extension est forcément présente au moment de la sauvegarde. Au final nous pensons avoir réussi à faire une interface assez aérée et agréable à utiliser mais jusqu'à maintenant nous avons parlé que du coté ergonomique de l'interface mais une autre majeure partie du travail a été concentrée sur l'aspect et le visuel de l'interface.

2.1.4 Apparence et thème CSS

L'ajout de ces nouvelles fonctionnalités a entraîné la création de nouveaux boutons et donc de nouvelles icônes !

Pour les nouvelles icônes nous sommes restés sur le même style graphique afin de garder une unicité dans l'application. Les icônes ont donc un format de 20x20 pixels sur fond transparents, toutes avec un même schéma de couleur : ce qui touche à l'image en dégradé de couleur (rouge et bleu) et le reste en noir et blanc.

Voici une liste exhaustive des nouveaux ajouts :

- Copier
- Coller
- Couper
- Flèche de rotation droite
- Flèche de rotation gauche
- Cacher/Montrer
- Pinceaux :
 - Carré
 - Etoile
 - Losange
 - Manuscrit
 - Rond
 - Trait horizontale
 - Trait verticale
- Position :
 - Nord-Ouest
 - Nord
 - Nord-Est
 - Ouest
 - Centre
 - Est
 - Sud-Ouest
 - Sud
 - Sud-Est

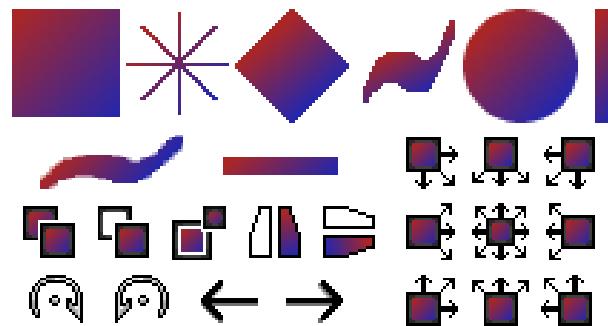


FIGURE 5 – Nouvelles icônes

-CSS

Pour la beauté de l'application et le confort de l'utilisateur nous avons créé plusieurs thèmes de couleurs. Nous avons donc créé 4 thèmes de base qui sont classiques mais néanmoins indispensables ! Et également 2 autres thèmes personnalisables par l'utilisateur dans l'application !

-Thème de base

Pour les thèmes de base nous retrouvons donc 4 thèmes ; Clair, Sombre, Clair contrasté, Sombre contrasté. Le thème **Clair** est directement inspiré du thème clair par défaut des applications gtk. Nous avons donc essayé de reproduire ce thème de base le plus fidèlement !

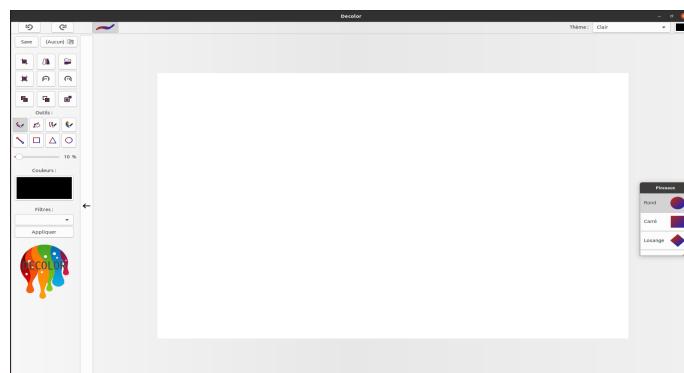
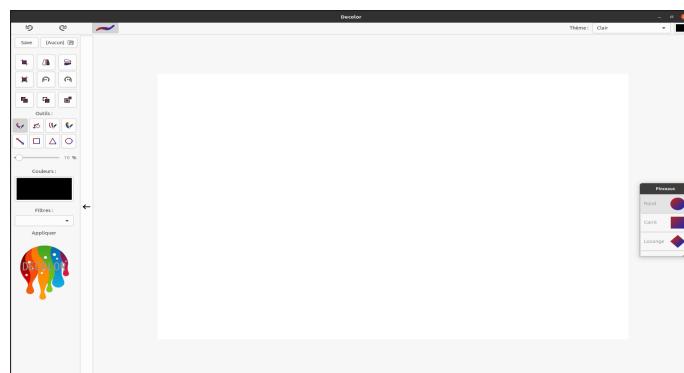


FIGURE 6 – Thème par défaut au dessus // Thème reproduit en dessous

Le thème **Sombre** est tout simplement à l'opposé du thème clair !

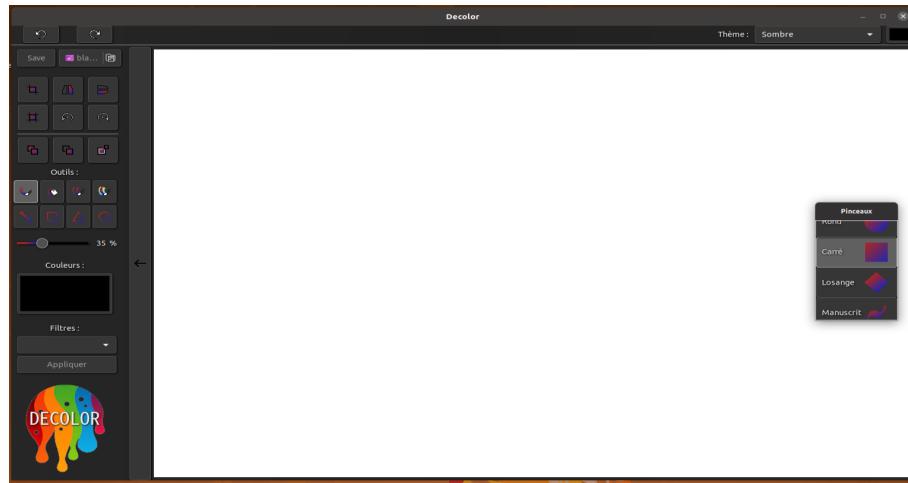


FIGURE 7 – Interface avec le thème sombre

Les thèmes **Contrastés** sont semblables aux thèmes normaux à la différence près que les différences entre les couleurs sont plus marquées, les boutons ressortent donc mieux avec des contours plus nettes.

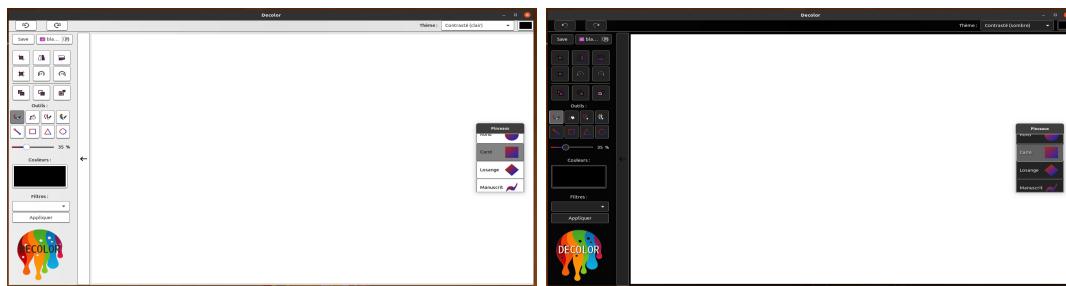


FIGURE 8 – Thème contrasté clair à gauche // Thème contrasté sombre à droite

- Thème personnalisé

Pour les thèmes personnalisés, c'est un poil plus complexe !

En effet, le principe est donc d'afficher un thème correspondant à la couleur sélectionnée par l'utilisateur ! Il faut donc dans un premier temps déterminer si la couleur choisie correspond à une couleur claire ou sombre. Pour cela il suffit de calculer son niveau de gris, de ce dernier nous pourrons déterminer quel fichier CSS utiliser et quels calculs appliquer.

Prenons un **vert clair**, pour exemple la couleur de code hexadécimal **#81F584**. Son niveau de gris est de **192** (#C5) cette couleur est au dessus du palier fixé, nous allons donc modifier et charger le fichier CSS correspondant aux couleurs claires.

Nous allons donc calculer 4 déclinaisons de la couleur, une couleur plus claire et 3 plus sombres, ainsi qu'un niveau de gris pour le texte :

Couleur = **#81F584** 
 Couleur plus claire = Couleur * 110 % + 10 = **#97FF9B** 
 Couleur plus sombre1 = Couleur * 90 % = **#74DC76** 
 Couleur plus sombre2 = Couleur * 80 % = **#67C469** 
 Couleur plus sombre3 = Couleur * 75 % = **#60B763** 
 Couleur texte = Niveau_de_gris(Couleur * 25 % - 10) = **#272727** 

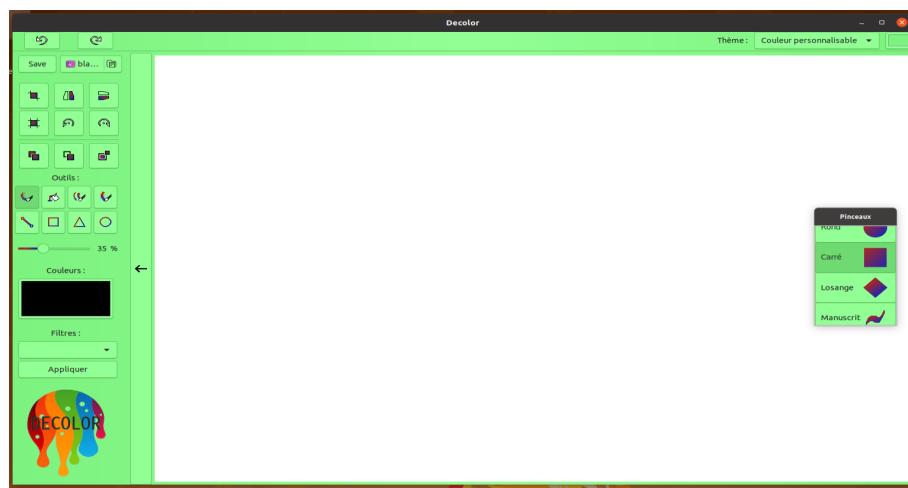


FIGURE 9 – Thème coloré clair

Prenons un **bleu foncé**, pour exemple la couleur de code hexadécimal **#091891**. Son niveau de gris est de **33** (#21) cette couleur est en dessous du palier fixé, nous allons donc modifier et charger le fichier CSS correspondant aux couleurs sombres.

Nous allons donc calculer 4 déclinaisons de la couleur, une couleur plus sombre et 3 plus claires, ainsi qu'un niveau de gris pour le texte :

Couleur = **#091891** 
 Couleur plus sombre = Couleur * 80 % = **#071274** 
 Couleur plus claire1 = Couleur * 300 % + 30 = **#3966FF** 
 Couleur plus claire2 = Couleur * 250 % + 15 = **#254BFF** 
 Couleur plus claire3 = Couleur * 110 % + 50 = **#3B4CD1** 
 Couleur texte = Niveau_de_gris(Couleur * 350 % + 140) = **#FEFEFE**

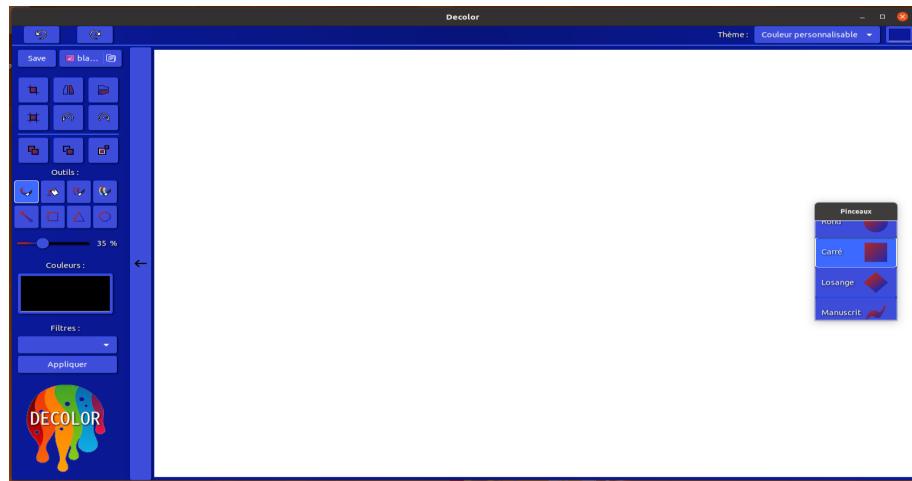


FIGURE 10 – Thème coloré sombre

L'utilisateur peut sélectionner les différents thèmes dans onglet se situant en haut à droite de l'interface, on y retrouve également une zone pour choisir la couleur du thème personnalisé.



FIGURE 11 – Zone de sélection du thème de l'interface

Pour aérer et embellir l'interface nous avons également rajouter un bouton permettant de rétracter la barre d'outils.

Quand bien même cette dernière est fermée, les options de retour en arrière, de retour en avant ainsi que de sélection des formes des pinceaux restent disponibles. Cela évite à l'utilisateur de devoir ouvrir et refermer l'onglet des outils trop souvent !

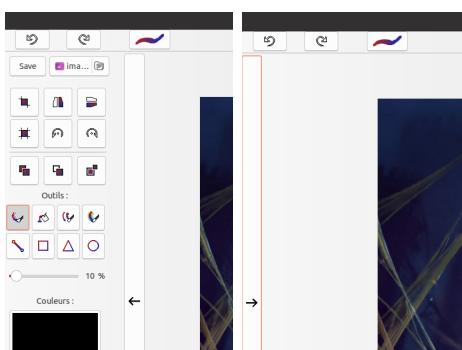


FIGURE 12 – Barre d'outils affichée // Barre d'outils rétractée

Par ailleurs le centrage de l'image a été retravaillé afin de prendre en compte plus de cas notamment celui du plein écran et de la rétraction de la barre d'outils

2.2 Nouveaux outils et modification des anciens

2.2.1 Le crayon

Tout d'abord, nous avons ajouté des crayons différents. Pour la 1ere soutenance, le crayon avait une forme de carré pas très belle à voir, alors nous avons d'abord modifié cette forme en disque puis nous avons créé beaucoup d'autres crayon que l'utilisateur peut sélectionner. On compte parmi ces crayons : le rond, le carré, le losange, le manuscrit (un crayon en forme de segment un peu penché pour la calligraphie), l'étoile, le trait vertical et le trait horizontal. Ces choix de crayon modifient aussi les formes !

Évidemment l'utilisation de chaque crayon varie en fonction de la taille et de la couleur sélectionnée.

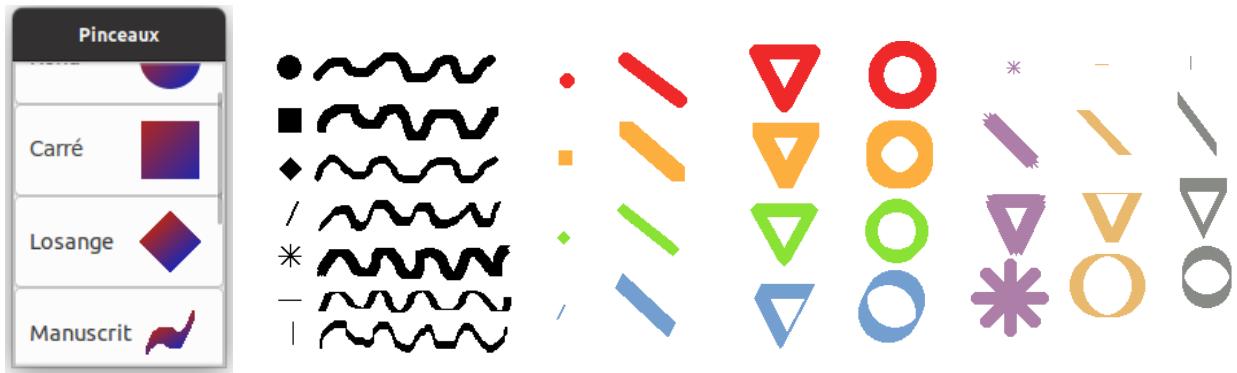


FIGURE 13 – Types de pinceaux, leurs tracés, et les différentes formes qu'ils génèrent

2.2.2 Les outils de transformations de l'image

Nous avions déjà codé l'inversion et le zoom pour la première soutenance mais ils n'étaient pas encore disponibles sur l'interface. Nous avons donc ajouté cet outil à l'interface en enlevant le zoom qui rendait l'image trop pixélisée et donc qui n'apportait aucune utilité au logiciel.



FIGURE 14 – Boutons Inversion

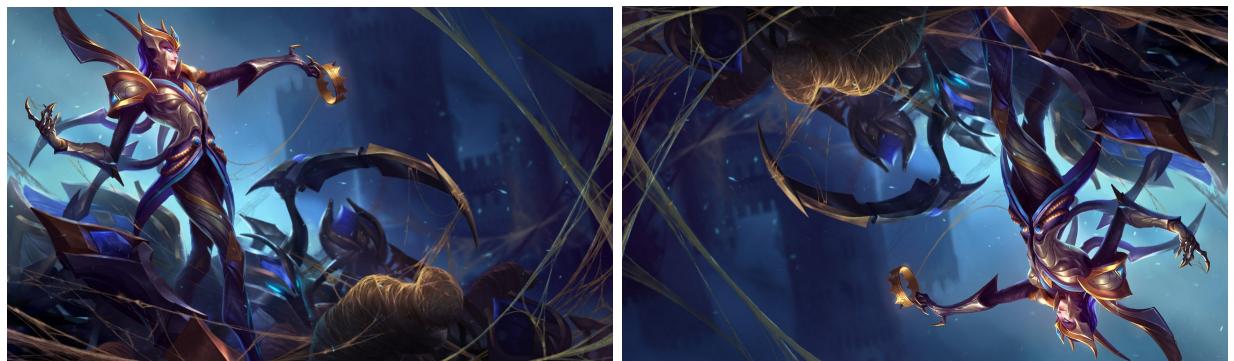


FIGURE 15 – Inversion de l'image horizontalement et verticalement

Nous avons ensuite ajouté quelques outils pratiques :

— **Rotations** :



FIGURE 16 – Boutons Rotation

Pour cette fonctionnalité, nous voulions tout d'abord utiliser la fonction d'une librairie directement disponible (`rotozoomSurface`), sauf qu'il s'est avéré qu'elle mettait un cadre noir non voulu aux bords de l'image. Le choix a donc été fait de créer nos deux propres fonctions pour tourner une image de 90 degrés vers la gauche et 90 degrés vers la droite. Des boutons ont ensuite été ajoutés sur l'interface pour pouvoir accéder à ces deux nouvelles fonctionnalités.

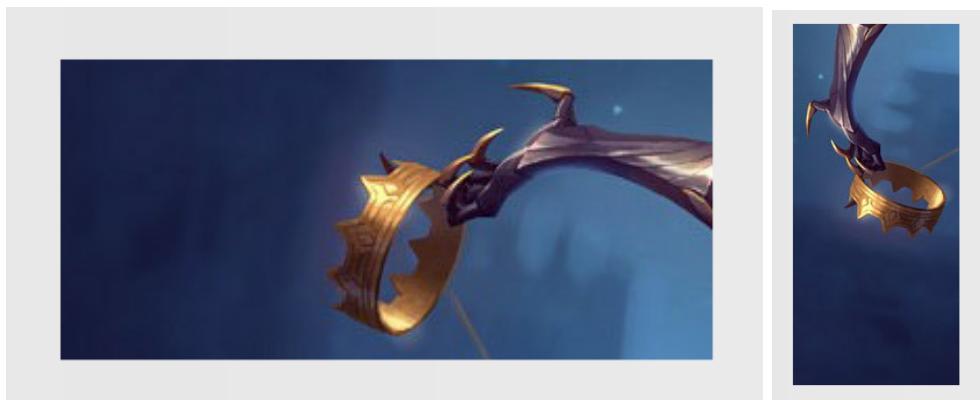


FIGURE 17 – Rotation de 90 degrés vers la gauche (Résultat rétréci pour rentrer dans la page)

— **Rogner l'image (Crop)** :

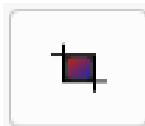


FIGURE 18 – Bouton Rogner

Afin d'intégrer cette nouvelle fonctionnalité clé de notre projet, nous avons utilisé les coordonnées de la souris au clic puis au relâchement du clic avec une vue en temps réel du cadre qui sera la nouvelle image avec un rectangle qui apparaît en tant que pré-visualisation. Une fois le clic relâché, une fonction s'occupe de récupérer des coordonnées valides pour l'image (les coordonnées pourraient être négatives ou dépasser le cadre de l'image) qui sont passés par la suite à notre fonction qui copie la partie de l'image sélectionnée dans une nouvelle image qui est ensuite affichée.



FIGURE 19 – Rétrécissement de l'image en fonction de la zone sélectionnée sur l'image de départ

— Redimensionner l'image :

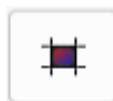


FIGURE 20 – Bouton Redimensionner

Cette nouvelle fonctionnalité a été un ajout capital pour la seconde soutenance. Une fois l'outil cliqué, comme précédemment expliqué dans la partie GTK, une fenêtre surgissante (pop-up) s'ouvre nous demandant les dimensions et le sens de la nouvelle image voulue. La fonction liée à cette fonctionnalité comporte donc 9 cas possibles. Il est aussi important de préciser que l'on peut, si souhaité, étendre seulement la hauteur et/ou la largeur de l'image comme rétrécir la hauteur et/ou la largeur de l'image en même temps.

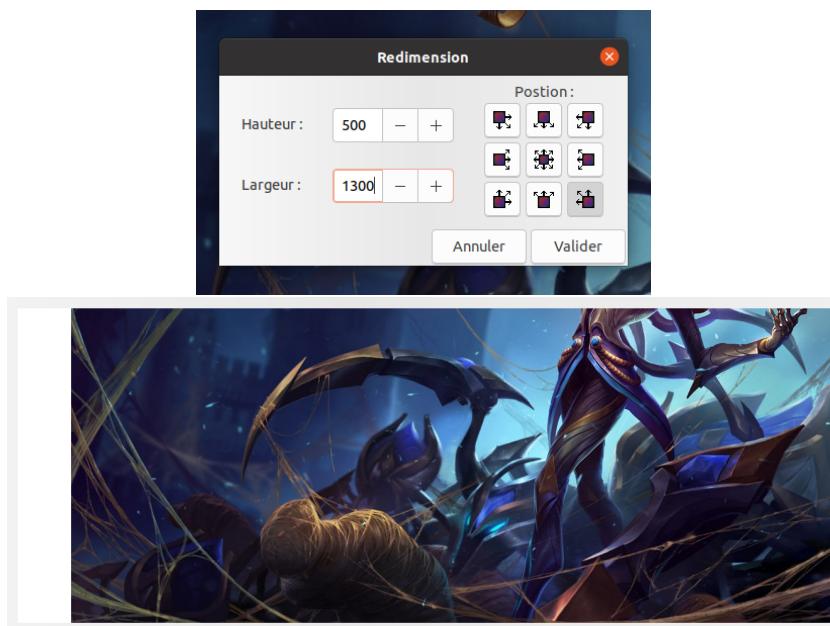


FIGURE 21 – Paramètres du redimensionnement de l'image voulu et résultat obtenu (Rétrécissement de la hauteur + agrandissement de la largeur + centrage de l'image de départ en bas à droite de la nouvelle)

— Copier :



FIGURE 22 – Bouton Copie

Pour continuer, nous avons ajouté un bouton afin de copier une zone sélectionnée de l'image. Une fois l'outil cliqué, nous devons juste sélectionner une partie de l'image que nous pouvons voir à l'aide de la pré-visualisation de la zone voulue. Par la suite une copie de cette zone sera enregistrée dans le logiciel et permettra à l'outil "Coller" de fonctionner correctement.

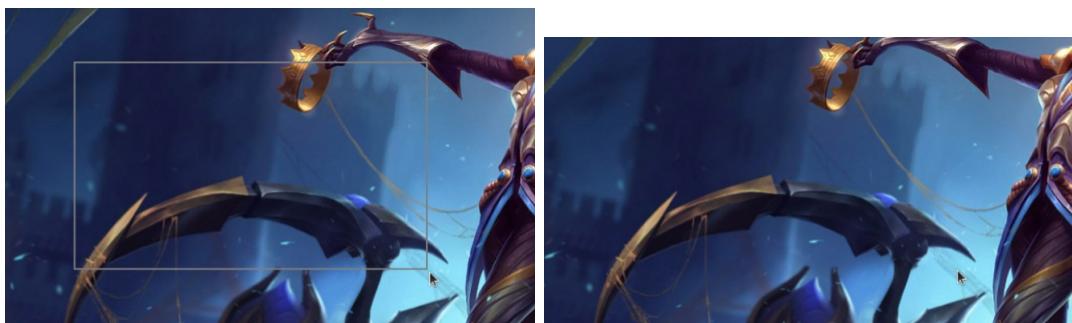


FIGURE 23 – La zone sélectionnée est copiée et aucun changement sur l'image de base n'a lieu

— Couper :



FIGURE 24 – Bouton Couper

Ce nouvel outil fonctionne en utilisant le même principe que l'outil "Copier". Cependant, à la différence de celui-ci, la zone sélectionnée est supprimée, c'est à dire qu'elle est effacée (remplacée par du blanc). L'exemple ci-dessous est utile pour mieux comprendre :

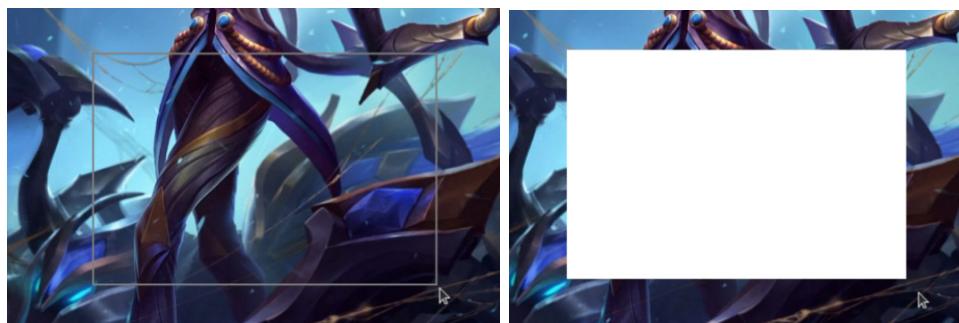


FIGURE 25 – La zone sélectionnée est copiée et est effacée sur l'image

— Coller :



FIGURE 26 – Bouton Coller

Cette nouvelle fonctionnalité est un complément des outils "Copier" et "Couper" qui lui permettent de fonctionner correctement. L'image qui est copiée et donc enregistrée dans le logiciel après l'utilisation de ces deux outils peut être posée sur l'image à n'importe quel moment et par 2 moyens différents. Tout d'abord, on retrouve le moyen du bouton qui appelle une fonction pour coller la copie n'importe où sur l'image à chaque pression du clic puis le second moyen consiste en l'appui en simultané des touches "ctrl" et "v" qui collera l'image copiée à l'endroit où se trouve la souris sur l'image.

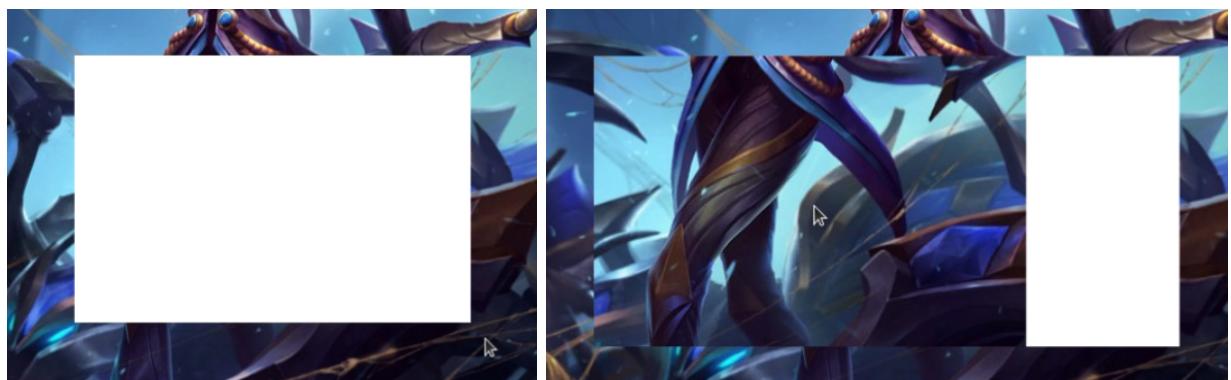


FIGURE 27 – On reprend l'image précédemment coupée, et on colle à coté (à gauche) ce qui a été copié

2.2.3 Les filtres

Nous avons ajouté beaucoup de filtres pour cette 2e soutenance car nous nous sommes rendu compte que c'est une façon simple et efficace d'améliorer notre logiciel. Cela ajoute un aspect "modification d'image" au logiciel car bien évidemment l'utilisateur peut superposer les filtres pour les combiner. Nous avions pour la 1e soutenance 3 filtres : nuance de gris, négatif et un filtre "factice" qui permettait de créer d'autres filtres sans avoir besoin de faire de calculs. Pour cette 2e soutenance nous nous sommes justement penchés dans les quelques calculs mathématiques sur les pixels que nous devions faire pour avoir des filtres jolis. Nous avons pu créer :

- **Le filtre luminosité** : il suffit d'ajouter un valeur n positive ou négative à chaque composante de chaque pixel (sans que ça ne dépasse 0 ou 255) pour rendre l'image plus claire ou plus sombre.

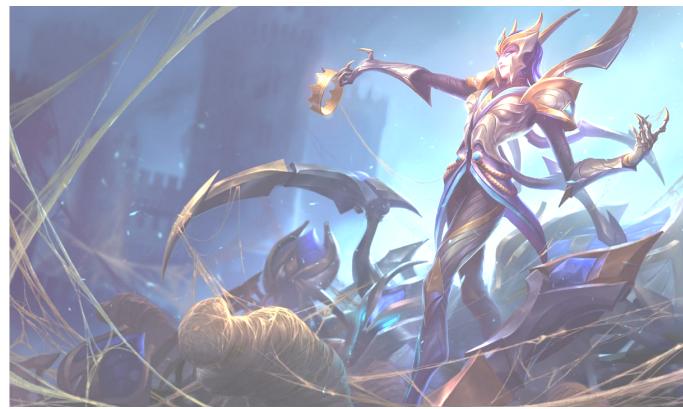


FIGURE 28 – Filtre luminosité avec $n = 50$

- **Le filtre contraste** : pour ce filtre, il faut rendre les pixels clairs encore plus clairs et les pixels sombres encore plus sombres en fonction d'un facteur donné. On applique donc une fonction qui se découpe en 2 parties : soit x la composante du pixel et n un facteur donné,

$$x < \frac{255}{2}, f(x) = \frac{255}{2} * \left(\frac{2x}{255}\right)^n$$

$$x > \frac{255}{2}, f(x) = 255 - \frac{255}{2} * \left(\frac{2(255-x)}{255}\right)^n$$

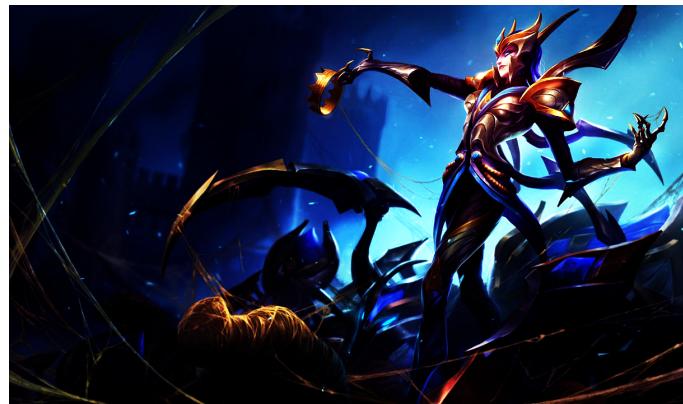


FIGURE 29 – Filtre contraste avec $n = 3$

- **Le filtre de couleur** : il suffit de faire la nuance de gris du pixel puis de la multiplier au pourcentage de chaque composante du pixel. C'est à dire que pour la composante rouge d'un pixel par exemple, soit rx, gx, bx les composantes d'une couleur x, alors la composante deviendra : $(r / 255 * \text{gris}, g / 255 * \text{gris}, b / 255 * \text{gris})$ avec $\text{gris} = 0.3*rx + 0.59*gx + 0.11*bx$

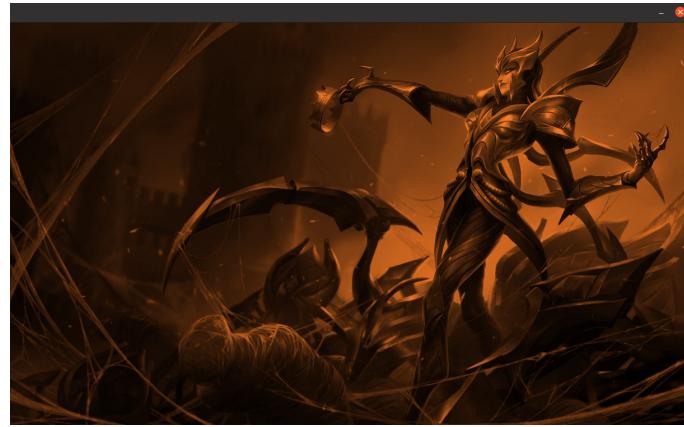


FIGURE 30 – Filtre obtenu en sélectionnant du orange

- **Le filtre flou** : nous avons dû utiliser le flou gaussien pour ce filtre. Le principe est simple, il faut, pour chaque pixel, faire la moyenne de tous les pixels à n distance de lui.

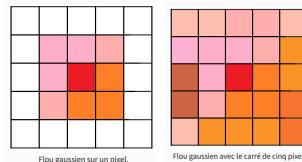


FIGURE 31 – Principe du flou gaussien sur un pixel puis sur cinq pixels

Si n est trop grand, le flou est évidemment très long à s'exécuter et on peut le remarquer sur n'importe quel logiciel, nous donc l'avons bloqué à 5.

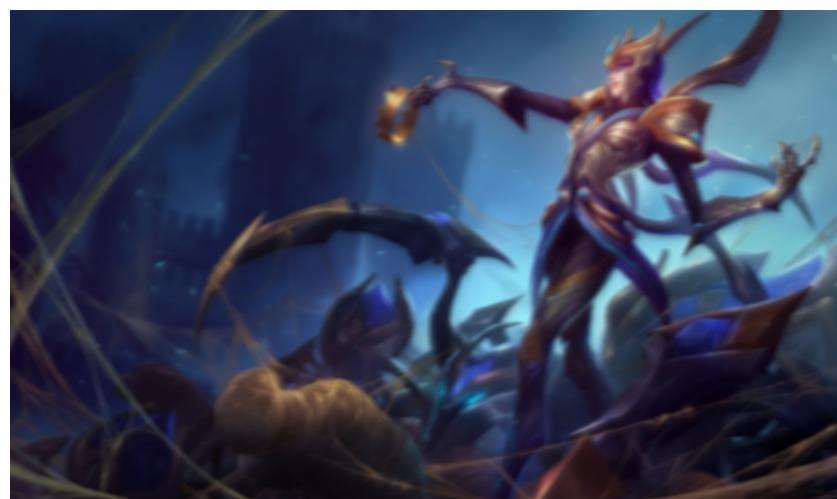


FIGURE 32 – Filtre de flou gaussien avec n = 5

- **Le filtre Détourage** : nous avons ici aussi utilisé le principe du flou gaussien pour ce filtre. Le principe est simple, il faut, pour chaque pixel, réaliser le calcul suivant :

$255 - | \text{la valeur du pixel} - \text{la moyenne de tous les pixels à } n \text{ distance de lui} |.$

Si n est trop grand, le détourage est évidemment très long à s'exécuter et on peut le remarquer sur n'importe quel logiciel, nous donc l'avons bloqué à 5.

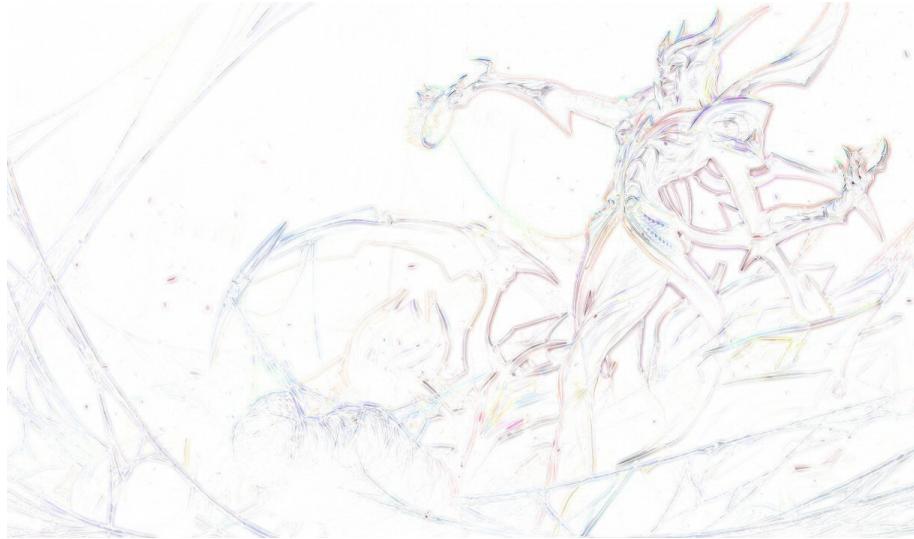


FIGURE 33 – Filtre de détourage avec $n = 5$

2.2.4 La pré-visualisation des formes et des outils

Afin de pouvoir mieux se repérer pour l'utilisation des formes 2D, nous avons mis en place un système qui montre en temps réel à l'utilisateur, tant qu'il maintient le clic de la souris, le rendu finale de l'image s'il relâchait le clic. Cet ajout est vraiment très utile notamment pour l'édition rapide et facile de l'image. Certains outils nécessitants la sélection d'une partie de l'image par l'utilisateur utilisent aussi la pré-visualisation qui prend la forme d'un fin rectangle gris. Avoir un aperçu de la zone sélectionnée permet d'éviter toute erreur et fait gagner en précision. On la retrouve pour les outils : "Rognage", "Copier" et "Couper".

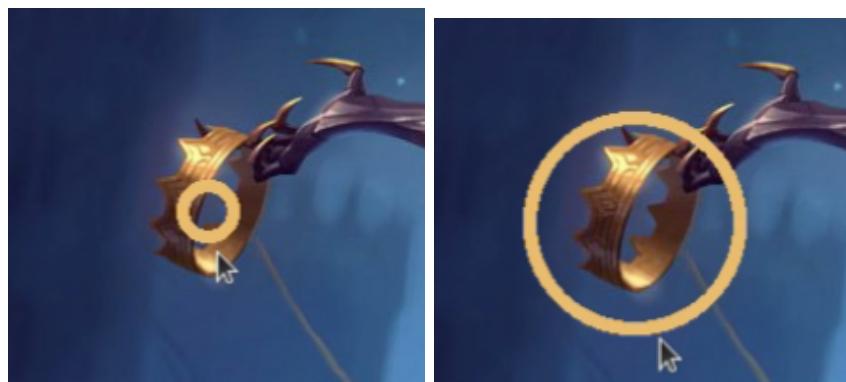


FIGURE 34 – Pré-visualisation (ici pour poser un cercle)

2.2.5 Fonctionnalité abandonnée

Pour cette 2e soutenance, nous avons voulu implémenter une grosse fonctionnalité sur notre logiciel : **les calques**. Nous avons créé une structure de liste chaînée qui permettait de lier chaque calque avec le précédent et une sentinelle au début de la liste qui compte tous les calques créés. Nous avions ainsi une structure fonctionnelle qui permettait de créer les fonctions basiques du calque simplement.

Malheureusement nous nous sommes rapidement rendu compte d'un gros problème d'optimisation : en effet, à cause des bibliothèques que nous utilisons pour le projet, la preview de ce que l'utilisateur est en train de faire aurait été difficilement possible car il aurait fallu 3 surfaces au lieu d'une (une pour les calques derrière, une pour celui sur lequel l'utilisateur dessine et une devant) et le crayon n'étant déjà pas parfaitement fluide avec une seule surface, nous ne pouvions pas nous permettre d'en ajouter 2 supplémentaires.

Nous avons donc abandonné cette partie du projet qui n'était de toute façon pas prévue à la base pour nous concentrer davantage sur l'amélioration des autres outils.

Nous avons aussi abandonné l'option de pouvoir **écrire du texte** sur nos images par manque de fonctionnalités sur SDL où nos images sont modifiées.

3 Site internet du logiciel

Le site Web de notre projet est avant tout le site Web de notre projet (OELA - Decolor). Ceux-ci sont respectivement représentés au travers de différents articles ainsi que liens de téléchargements et autres interactions dans le but d'attirer les visiteurs à télécharger l'application.

La structure du site Web est entièrement terminée mais sera mise à jour suite aux nouveaux ajouts.

Voici le lien pour aller le consulter, https://topagrume.github.io/decolor_web/accueil.html

Possédant déjà un compte *GitHub*, nous nous sommes tournés vers ce support pour héberger le site et le mettre à jour automatiquement au fil des nouvelles versions envoyées vers la sauvegarde distante. A l'aide de nos connaissances en HTML (pour la mise en forme des pages), CSS (pour le style et l'apparence des pages) et JavaScript (pour ce qui est du code à exécuter côté client surtout les effets dans notre cas), des forums en ligne ainsi que d'exemple d'organisation, de répartitions du texte, d'effets et d'autres outils disponibles librement sur Bootstrap et d'autres espaces de libre-échange, nous avons pu réaliser ce site.

Nous nous sommes basés sur le thème de notre logo Decolor pour l'apparence du site qui allie des couleurs vives. Le langage CSS n'étant pas toujours évident, cela a été passé par beaucoup de tâtonnements, de découvertes puis aussi d'exploration de documents sur Internet, mais nous sommes aujourd'hui très satisfaits du résultat qui s'adapte de plus à de nombreuses tailles d'écran mis à part l'arbre d'historique de notre projet avec les écrans de téléphones. La navigation est alors agréable, intuitive et plus aisée.

3.1 La structure du site

Il a tout d'abord fallu réfléchir à la structure du site, son arborescence, l'organisation de chacune des pages, la répartition des images, du texte, des liens, d'un historique de projet, de texte et de statistiques. Nous avons décidé de choisir une structure simple composée d'un en-tête dans lequel on retrouve le nom de notre groupe, le nom et le lien vers les différentes pages du site : (Projet, Avancées, Documentation, Équipe) ainsi qu'un bouton « Télécharger ». Puis dans la suite de la structure un corps de texte qui laisse libre recours à nos choix et explications et enfin un pied de page destiné à n'être qu'un simple décor.

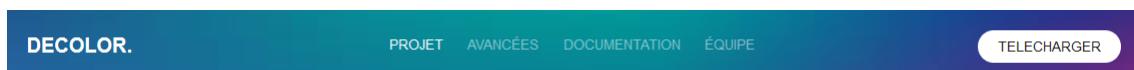


FIGURE 35 – Structure du site

3.2 Page d'accueil

Bien en évidence, nous avons opté pour directement placer un bouton afin de renvoyer vers un lien de téléchargement du logiciel en version normale/lite. Dans le corps de la page, nous avons opté pour une présentation brève de notre logiciel, et de ses qualités, répartie en différents onglets mis en parallèles avec des images représentatives, puis de nos valeurs en tant que développeur, une présentation brève de l'intérêt algorithmique et de son fonctionnement et enfin un petit paragraphe sur notre politique ainsi que le choix de notre Logo.

Plus bas, il nous a semblé important présenter les différents points d'avancement par des images représentatives explicites.

Enfin en pied de page, nous avons implémenté des statiques à titre purement informatif et esthétique sur le nombre de nos commit, d'ajouts réussis, d'étapes réussies et enfin le nombre d'heures qui ont été dédiées au projet dans son ensemble.

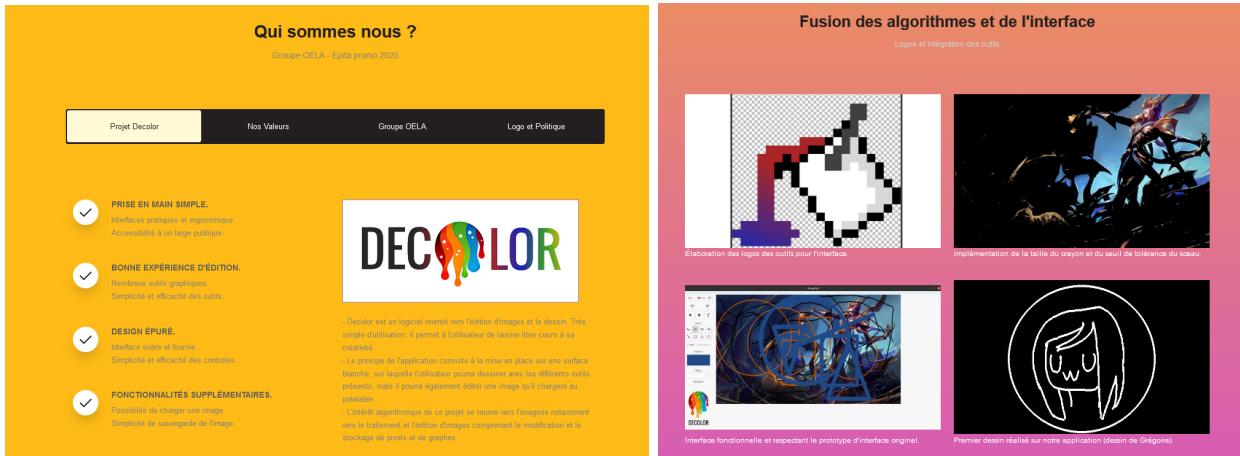


FIGURE 36 – Présentation du projet et des ajouts

3.3 Page d'avancements

Nous avons ici opté pour présenter les différentes étapes clés du projet passées tout au long de notre semestre (mars à juin) avec notamment de nombreux détails sur l'état de notre avancée, de nos choix, de nos réorientations, de notre productivité, de nos implémentations et étapes passées.

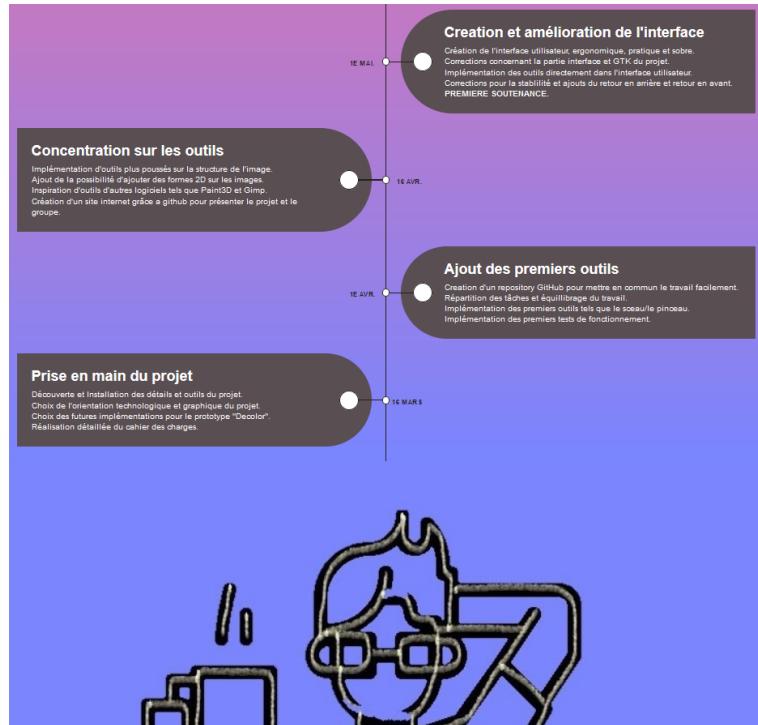


FIGURE 37 – Page d'avancements en fonction du calendrier

3.4 Page de documentation

Dans cette partie nous avons rassemblé l'ensemble des documents qui nous ont été demandés durant l'avancement de notre projet à savoir notre cahier des charges, première et deuxième soutenance. Elles

sont présentées avec leurs différentes caractéristiques avec un bouton pour les télécharger.

Dans la lignée des présentations, nous avons rassemblé l'ensemble des logiciels et des outils qui nous ont permis de faire aboutir notre projet.

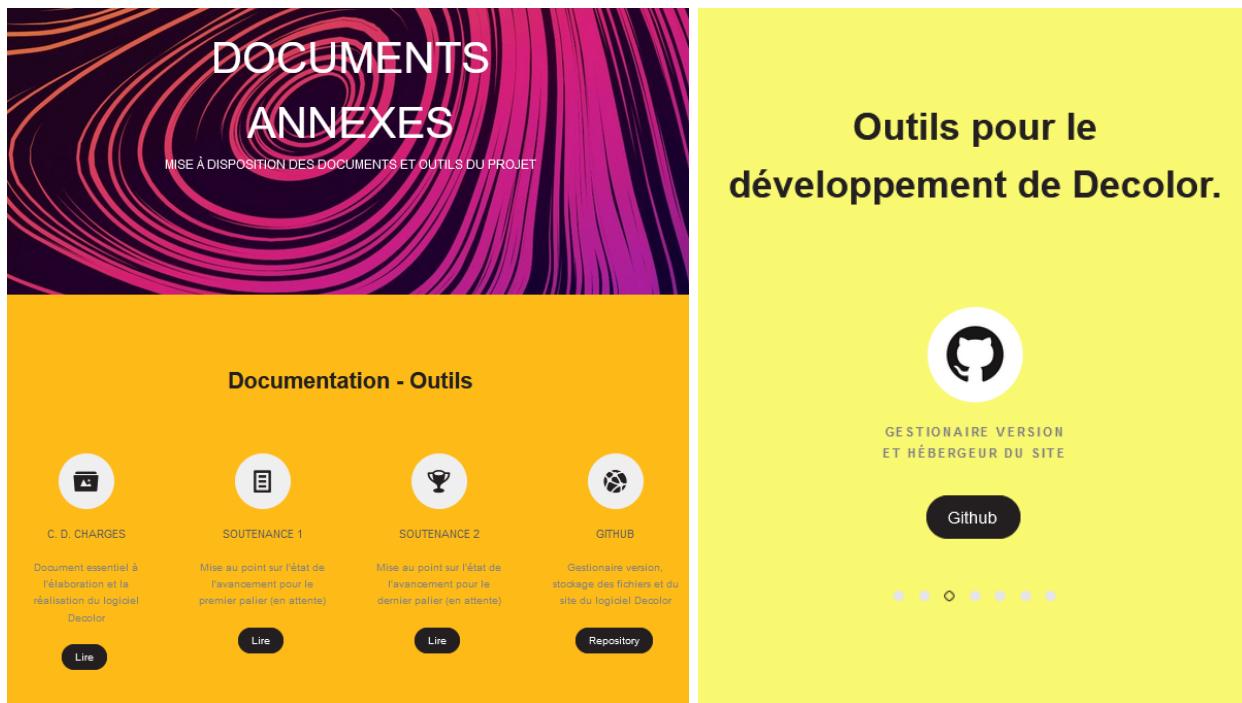


FIGURE 38 – Page documentations et outils

3.5 Page de l'équipe

Il nous a semblé important de présenter les différents membres du projet individuellement avec notamment leurs domaines de travail. Les liens vers le Twitter et le GitHub de chaque membre est renseigné sous chaque personne. Nous avons aussi simplement disposé quatre moyens pour pouvoir nous contacter ou qui concernent la partie communication de notre projet.



FIGURE 39 – Présentation de l'équipe

3.6 Page de téléchargement

Lorsque le bouton télécharger est activé, un ".tar.gz" correspondant à la version choisie de notre projet est téléchargé. Après l'avoir décompressé, il est fonctionnel. Une version du projet en version lite (sans fichiers de test/inutiles) et une version normale sont actuellement disponibles au téléchargement.

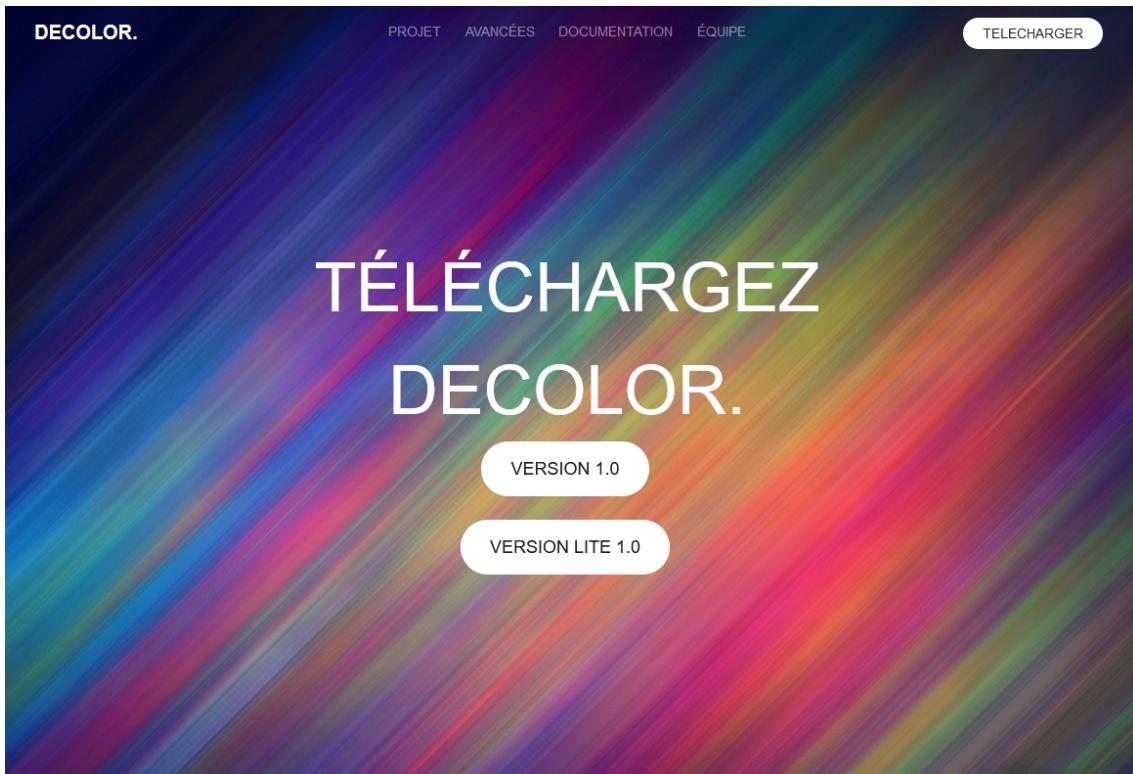


FIGURE 40 – Page de téléchargement

4 Découpage du projet

4.1 Tableau de répartition des tâches

	Alexandre	Kaël	Grégoire	Mathis
GTK - Interface		Supp		Resp
Interface graphique globale		X		*
Implémentation des outils	*	X	*	*
Chargement et enregistrement des fichiers				X
Implémentation de l'environnement de travail	*	*		X
Fenêtres informatives		*		X
SDL	CoResp		CoResp	
Création des outils	X		*	
Création des formes 2D	X		*	
Gestion des images	*		X	
Édition de la structure des images	X	*	*	
Application de filtres		*	X	*
Structure du Projet	Supp	Resp		
Ergonomie de l'application	*	X		
Compilation des fichiers (Makefile, headers...)	X	*	*	*
Communication	Resp	Supp		
Site Web	X	*		
Rendus Overleaf	X	*	*	*

Légende :

X = Responsable, * = Suppléant

4.2 Prévisions soutenances

Soutenance Tâche	Prévisions Soutenance 2	Accompli Soutenance 2
GTK - Interface	100	110
SDL	100	100
Structure du Projet	100	100
Communication	100	100

TABLE 1 – Tableau des soutenances (en %)

5 Développement sur les tâches réparties et le travail

5.1 Schémas Github - Contributions de chacun

Pour une raison non connue, les contributions en lignes ajoutées et supprimées de Grégoire n'apparaissent pas. Cependant il est possible de voir son nombre de commit dans la deuxième photo (Il est en 3e position).

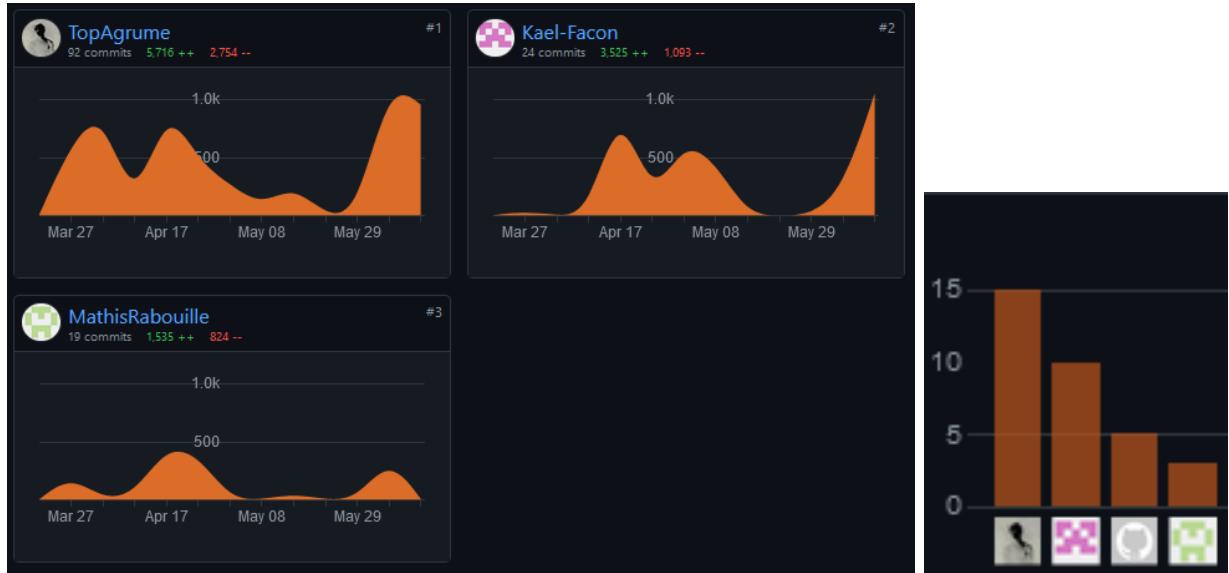


FIGURE 41 – Contribution GitHub avec respectivement Alexandre, Kaël, Grégoire et Mathis (2e image)

5.2 alexandre.devaux-riviere

Lors de cette seconde soutenance, je me suis beaucoup investi dans la création des outils et des implémentations de ceux-ci dans l'interface. Je me suis notamment occupé de la pré-visualisation des outils / formes 2D, des rotations, du rognage, de la redimension des images, du filtre "détourage", et des outils "Copier", "Couper" et "Coller". Je me suis de même occupé de créer des tracés pour améliorer le crayon.

La cohésion du groupe, aidée par l'amitié qui nous liait déjà, a fait de ce projet une réussite. Nous avons passé de longues soirées à travailler "tous ensemble" sur le projet, en échangeant avis et conseils. Le fait d'apprendre de nouvelles choses tout en créant a été d'une grande motivation durant tout ce projet. Ma méthode de développement s'est en partie basée sur mon expérience de travail en tant que utilisateur de ce type de logiciel.

Personnellement, je considère ce projet comme une expérience très enrichissante que ce soit au niveau des notions de travail en groupe tout comme pour les connaissances en C que nous avons pu acquérir au cours de cette période.

5.3 kael.facon

Lors de cette deuxième et dernière soutenance mon objectif principale était d'implémenter le maximum d'options dans l'interface tout en essayant de la laisser aérée et sans réduire l'espace de travail disponible pour l'image. Je pense avoir plutôt atteint cet objectif grâce aux fenêtres flottantes et surgissantes ainsi

qu'à la barre d'outils rétractable.

L'implémentation de ces nouvelles fonctionnalités m'ont permis d'étendre mes connaissances vis-à-vis de la librairie GTK notamment la création de fenêtres via le code et non via le logiciel glade ainsi que leur suppression. J'ai également illustré nos outils avec des icônes dans un seul et unique style afin de garder une cohérence pour le logiciel. Nous avons maintenant une identité visuel qui se démarque et que tout le monde peut reconnaître au premier regard.

Je suis content de la tournure qu'a pris ce projet, grâce à un groupe motivé et inspiré pour travailler nous avons atteint nos objectifs et sommes même allés au delà ! Notre travail était coordonné et la communication était bonne !

5.4 gregoire2.vest

Pour cette soutenance, j'ai cherché des idées d'outils à implémenter. Tout d'abord j'ai ajouté quelques filtres jolis qui rendent le logiciel plus intéressant, puis j'ai cherché quelques crayons différents à implémenter et enfin j'ai voulu rendre disponible l'utilisation des calques. Malheureusement nous n'avons pas pu finaliser cette dernière partie car les bibliothèques que nous utilisons ne le permettaient pas.

5.5 mathis.rabouille

Ma partie du travail a concerné GTK pour l'implémentations des outils et le travail de l'ergonomie de l'interface. Ce travail ne m'a pas pris énormément de temps et là pour la majorité de cette soutenance j'ai travaillé comme assistant pour aider les autres à régler leurs problèmes. Mon travail a été assez découpé car il y a eu les partiels au milieu de cette soutenance mais au global je sens que le projet et l'interface sont satisfaisants.

6 Nous contacter

- Alexandre Devaux-Rivière : alexandre.devaux-riviere@epita.fr
- Kaël : kael.facon@epita.fr
- Grégoire: gregoire2.vest@epita.fr
- Mathis : mathis.rabouille@epita.fr