

SOUTENANCE 3

2021-2022



Team ALPH



Léo Devin B2

Christophe Nguyen C1

Phu Hung Dang B2

Angélo Eap B2

Table des matières

1	Cahier des charges	4
1.1	Choix du logo ALPH	4
1.2	Prototype de logo	4
1.3	Origine	6
1.4	But, intérêt algorithmique, principe	6
1.5	État de l'art	7
1.6	Inspirations artistiques et fonctionnelles	7
1.7	Objet de l'étude	8
1.8	Fonctionnalités	9
1.8.1	Les outils	9
1.8.2	L'édition	11
1.9	Lancement de l'application	13
1.10	Moyens matériels	13
1.11	Moyens intellectuels	13
2	ALPH	14
2.1	Notre Groupe	14
2.2	Le Projet	14
3	Choix des bibliothèques	14
3.1	Simple DirectMedia Layer - SDL	14
3.2	GIMP Toolkit - GTK	15
3.3	Cairo - GTK	15
3.4	GNU Compiler Collection - GCC	15
3.5	Makefile - MAKE	15
4	Notre Projet - Pinte	15
4.1	Première aperçu de notre projet	15
4.1.1	Interface	15
4.2	Les outils	17
4.2.1	Formes	22
4.2.2	Formats de la page	27
4.2.3	Filtres	35
4.2.4	Conversion de l'image pour les filtres	37
4.2.5	Rotation	38
4.2.6	Système de sauvegarde de fichier	38

4.2.7	Système de sauvegarde automatique	39
4.2.8	Outils en cours d'utilisation	39
4.3	Lancement de l'application	40
5	Site internet	40
5.1	La structure du site :	41
5.1.1	La page d'accueil :	41
5.1.2	La page de présentation de l'application	41
5.1.3	La page des documents :	42
5.1.4	La page à propos :	42
5.1.5	La page contact :	42
5.2	La réalisation du site	42
5.3	L'apparence du site	43
6	Les différentes tâches	47
6.1	Travail personnel et ressenti du projet	48
6.1.1	Léo DEVIN	48
6.1.2	Christophe NGUYEN	49
6.1.3	Phu Hung DANG	50
6.1.4	Angélo EAP	51
6.2	Tableau de répartition des tâches	52
6.3	Visualisation de l'historique de développement	53
7	Problème rencontrés et solution trouvées	54
8	Avance et Retard	55
9	Avis et commentaires du groupe	55
10	Anecdotes	55
11	Conclusion	55

1 Cahier des charges

1.1 Choix du logo ALPH

Nous avons opté pour un logo simple, discret mais aussi à la fois efficace et moderne regroupant l'ensemble des lettres de notre nom de groupe.



1.2 Prototype de logo

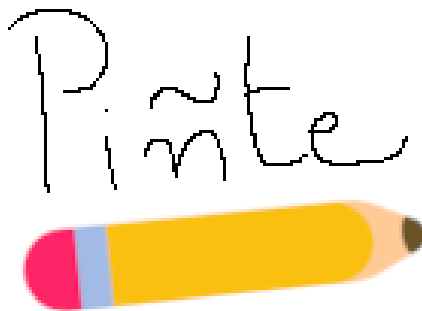


FIGURE 1 – prototype 1



FIGURE 2 – Prototype 2



FIGURE 3 – Prototype retenu

Notre logo n'est encore qu'un prototype et peut, de ce fait, complètement changer, mais il concilie jusqu'ici parfaitement la thématique de l'application que nous vous présentons ci-dessous.

1.3 Origine

Chaque membre du groupe étant très familier avec les bibliothèques SDL et GTK, nous avons décidé de réaliser une application similaire à celle du projet de S3.

Notre application utilisera GTK comme bibliothèque graphique pour réaliser notre interface, cela nous permettra d'implémenter par exemple des boutons pour appeler certaines fonctions ou montrer une image.

La bibliothèque SDL sera aussi utilisée pour la manipulation d'images ce qui nous permettra de récupérer une image sous forme de "surface" que l'on pourra stocker en mémoire de manière à pouvoir la modifier à notre guise. Nous avons pu manipuler ces bibliothèques lors de notre projet précédent, ce qui pourra nous être très utile.

L'application choisie est inspirée de plusieurs applications d'édition d'image et de dessin tel que Paint, qui existe sur Windows et non sur Linux.

Le type d'application choisie n'a pas été direct, nous nous sommes d'abord rassemblés et avons débattu sur plusieurs idées et thèmes ayant du potentiel.

1.4 But, intérêt algorithmique, principe

L'application, nommée Piñte, 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 aura une fonction qui permettra de charger une image à partir d'un explorateur de fichier et de venir dessiner sur cette dernière ou de la manipuler. Par la suite, cette image pourra être sauvegardée dans les fichiers locaux de l'ordinateur. Nous allons opter pour une interface sobre et simple 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.

Le principe de l'application est de mettre l'utilisateur directement sur une surface blanche, sur laquelle il pourra dessiner avec les différents outils présents, mais il pourra également éditer une image qu'il chargera au préalable.

Ce projet nous permettra donc d'acquérir de l'expérience que ce soit en programmation, en designing ou encore en communication.

1.5 État de l'art

Bien qu'il existe déjà des applications ressemblant à Paint sous linux telle que TUX Paint, Gpaint ou GIMP notre application aura pour but de simplifier au maximum l'expérience d'édition de l'image tout en gardant son efficacité. Pour cela, nous simplifions au maximum l'interface graphique ainsi que les différents outils qui seront présents.

Notre interface doit cependant comprendre assez d'éléments ainsi qu'un style assez sobre, donc un juste milieu entre une application pour enfant et une application de professionnel.

1.6 Inspirations artistiques et fonctionnelles

Pour la ligne directrice de notre projet, nous nous sommes inspirés de :

GIMP : Pour le outils

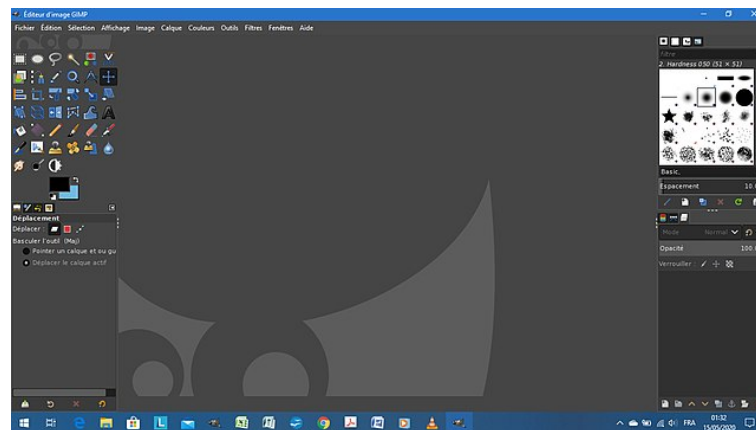


FIGURE 4 – GIMP

Points forts :

- Le nombre important d'outils.

Points faibles :

- Application peu intuitive

Paint : Pour les graphismes

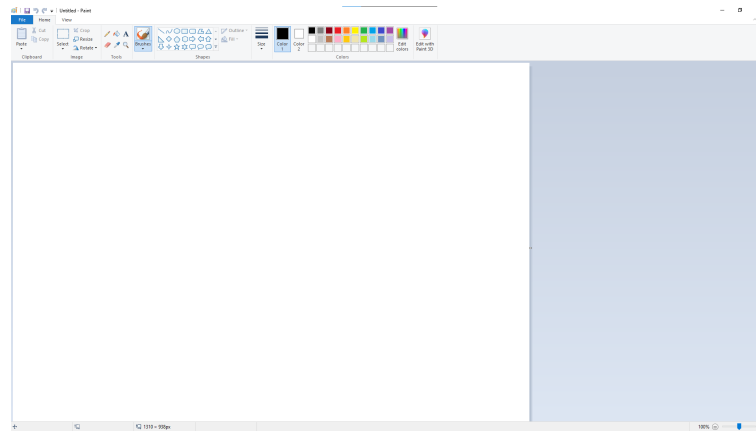


FIGURE 5 – Paint

Points forts :

- Simplicité et efficacité des graphismes et de l'application.

Points faibles :

- Manque d'outils professionnel.
- Non gestion de la transparence.

1.7 Objet de l'étude

La création de cette application nous permettra d'approfondir nos compétences en C, SDL et GTK et sur différents outils, tels que *Overleaf* pour les rendus et *GitHub* pour sauvegarder notre progression. De plus, celui-ci nous donnera une autre belle expérience de travail en groupe, indispensable à tout bon ingénieur. Dans notre vie professionnelle nous serons aussi tous amenés à faire des projets et à travailler en groupe, alors nous entraîner davantage possible est un vrai avantage.

1.8 Fonctionnalités

1.8.1 Les outils

Différents outils seront présents dans l'application permettant une manipulation simple d'images.

Crayon

L'outil crayon servira à l'utilisateur de se mettre en mode dessin. Lorsqu'il appuiera sur le clique gauche de sa souris il pourra faire des traits de la couleur de son choix, sélectionnable à partir de la palette de couleurs. Il pourra de plus changer à tout moment l'épaisseur de son crayon.

Palette de couleurs

De plus, nous ajouterons une large palette de couleurs permettant à l'utilisateur d'avoir un large choix mais aussi de pouvoir choisir une couleur à partir des valeurs de rouge, vert et bleu, lui permettant ainsi d'avoir accès à une infinité de couleurs.

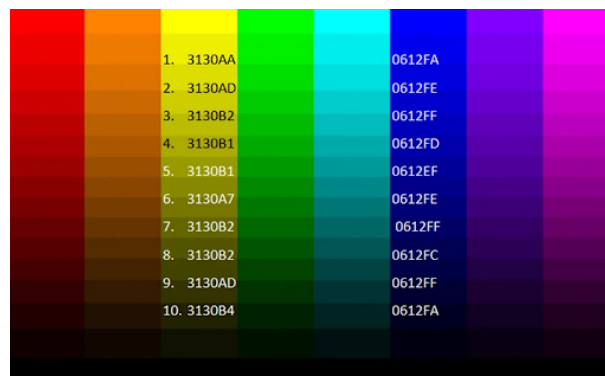


FIGURE 6 – palette de couleurs

Seau

Le seau permettra le remplissage d'une zone avec la couleur sélectionnée.

Forme

nous souhaitons implémenter un outil permettant de créer des formes géométrique simples telles qu'un cercle, un carre, un rectangle. Ces formes seront créées a l'aide d'algorithmes, qui a partir des coordonne du pointeur, créera la forme géométrique d'une taille variable.

Pipette

L'outil pipette est un outil très utile et simple d'utilisation. En effet il s'agit de récupérer une couleur présente sur l'écran. L'utilisateur sélectionnera l'outil, puis il sera libre de choisir une couleur parmi tous les pixels présents sur l'écran, il appuiera ensuite sur le clic gauche de la souris, pour confirmer son choix et la couleur sera présente dans la palette de couleur.

Loupe

Afin de faciliter l'ergonomie dans notre application, nous ajouterons un outil loupe qui permettra d'améliorer la précision dans les tracés. L'outil loupe est tout simplement un outil permettant de d'agrandir la partie d'une image souhaitée. Lorsque l'utilisateur sélectionnera l'outil, il pourra sélectionner un endroit sur son image, puis après un clic gauche, la zone s'agrandira sur la fenêtre.

Cet outil sera très utile pour corriger les défauts minimes afin de perfectionner le dessin. Ainsi, la loupe est un outil indispensable dans la conception de notre projet. Esthétiquement, la loupe sera positionnée dans un coin supérieur ou inférieur de l'écran de l'utilisateur. L'outil loupe ne gênera pas l'utilisateur quand il dessine et il aura l'option de la désactiver s'il le souhaite.

1.8.2 L'édition

L'édition permettra de modifier une image comme par exemple l'agrandir, la rétrécir, lui appliquer une rotation ou un filtre.

Rotation

Une option de rotation sera disponible afin de pivoter les images ou les sélections. Concernant les images, nous pourrons les faire tourner grâce à des touches du clavier ou bien la souris. Les dessins réalisés par l'utilisateur devront d'abord être sélectionnés grâce à un outil "sélectionner", puis ils pourront être pivoter.

L'outil de rotation est omniprésent dans toutes les applications ou logiciels traitant le dessin ou les images, c'est pourquoi cet outil est indispensable dans la réalisation de ce projet.

Filtre

Grâce au filtre, l'utilisateur peut changer le visuel de l'image. Par défaut, notre application propose différents filtres comme une image en niveau de gris, une image plus contrastée, une image plus lumineuse, une image avec un niveau de rouge plus important, etc...



FIGURE 7 – Filtres à disposition

Rogner

L'utilisateur pourra rapidement modifier la forme d'une image en la rognant pour lui donner une forme spécifique qu'il souhaite avoir. L'image est automatiquement découpée pour remplir la géométrie de la forme sélectionnée tout en conservant ses proportions.

Lorsque vous souhaitez rogner une image, des traits épais noirs apparaissent aux coins de l'image et au milieu des différents côtés. Le rendu de la nouvelle image sera les pixels qui seront contenus dans les traits noirs.

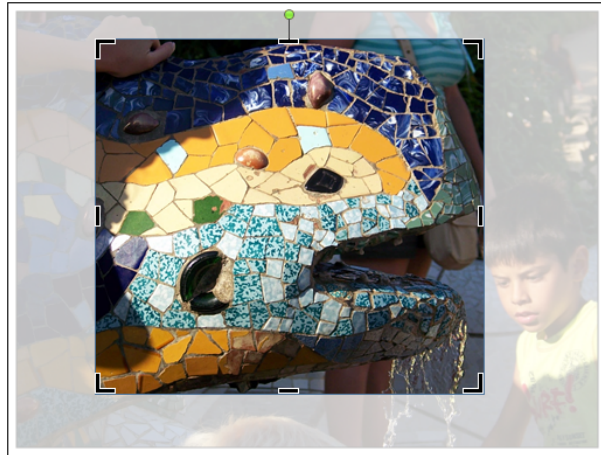


FIGURE 8 – Rogner une image

Agrandir et rétrécir

Si l'utilisateur souhaite agrandir une partie de son dessin, il aura à disposition un outil permettant d'agrandir ou de rétrécir les images. De plus, l'utilisateur n'a pas forcément le contrôle de la taille des images qu'il importe depuis Google Images, c'est pourquoi cet outil sera très utile.

La complexité dans la réalisation de cet option sera sûrement de garder la bonne proportion de l'image sans la déformer.

Transparence

L'option de transparence est indispensable pour une application de traitement d'images.

1.9 Lancement de l'application

Création

Si l'utilisateur souhaite seulement dessiner, il peut créer une page vierge blanche qui aura comme dimension les mesures qu'il souhaite avoir.

Chargement de l'image

Si l'utilisateur veut éditer une image, il doit charger une image au début de son travail. Pour charger une image, il y aura un bouton permettant de choisir une image depuis son ordinateur. L'image pourra ainsi être modifiée depuis notre application.

Édition

Lorsque l'utilisateur aura chargé son image à partir de ses fichiers, il la verra sur notre surface de départ, prête à subir des modifications. En effet, elle pourra s'agrandir, se rétrécir, être rognée, on pourra dessiner par dessus, mettre des formes et appliquer les filtres disponibles .

1.10 Moyens matériels

Nous avons accès grâce à l'EPITA à une vaste variété de matériaux et de logiciels pouvant nous aider à développer et concrétiser notre projet. Nous avons accès aux locaux de l'EPITA ainsi qu'à nos propres machines, et *Google* pour nos recherches. Nous allons également utiliser *VIM* ainsi que *EMACS* pour coder et pour finir, tout support papier sera conçu en \LaTeX .

1.11 Moyens intellectuels

Nous avons à disposition Internet qui est une grande bibliothèque contenant les informations nécessaires nous expliquant comment utiliser les logiciels mis à disposition ou développer dans certains langages à travers les différents tutoriels, manuels numériques sur les logiciels et forums d'entraide existant. Nous aurons également la possibilité de recevoir de l'aide de la part des professeurs de l'EPITA, des ASMs et des étudiants des années supérieures de l'école, qui ont tous plus d'expérience que nous.

2 ALPH

2.1 Notre Groupe

- Le groupe est composé de :
- Léo Devin (Chef de projet)
 - Christophe Nguyen
 - Phu Hung Dang
 - Angelo Eap

Notre groupe, nommé ALPH, fondé le 9 janvier 2022. Composé de 4 personnes dont les idées convergeaient, notre objectif est de réaliser un outil graphique permettant aux utilisateur sous Linux de profiter pleinement d'une application d'édition d'image matricielle, mais qui peut également servir de tableau, ce qui permettra aux artistes de déployer leur plein talent de dessinateurs accomplis.

2.2 Le Projet

L'application, nommée Piñte, 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 aura une fonction qui permettra de charger une image à partir d'un explorateur de fichier et de venir dessiner sur cette dernière ou de la manipuler. Par la suite, cette image pourra être sauvegardée dans les fichiers locaux de l'ordinateur.

3 Choix des bibliothèques

3.1 Simple DirectMedia Layer - SDL

SDL est une bibliothèque logicielle libre qui nous permet de travailler sur la manipulation d'images. Nous déjà pu utiliser cette bibliothèque lors de différents TPs ou encore lors de notre projet de troisième semestre qui consistait à résoudre un sudoku.

3.2 GIMP Toolkit - GTK

GTK est une aussi bibliothèque logicielle permettant de réaliser des interfaces graphiques. Nous avons opté pour GTK car nous avons aussi déjà pu l'utiliser. Il 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.

3.3 Cairo - GTK

Cairo est la bibliothèque utilisé par GTK pour le dessin vectoriel. Il nous est donc utile pour notre fonction dessin de notre application.

3.4 GNU Compiler Collection - GCC

Nous avons décidé d'utiliser GCC afin de compiler notre projet, codé en C car c'est le principal compilateur que l'on utilise en cours.

3.5 Makefile - MAKE

Make nous permettra de construire automatiquement nos fichiers exécutables avec les bibliothèques que nous utilisons. Un Makefile sera présent dans les fichiers pour pouvoir compiler rapidement notre projet.

4 Notre Projet - Pinte

4.1 Première aperçu de notre projet

4.1.1 Interface

Notre interface est composée sur la droite d'une grande surface permettant à l'utilisateur de dessiner et d'afficher son image, et sur la gauche tout les outils lui permettant de mener à bien son travail. Son thème est assez sobre, simple, permettant une prise en main rapide. L'interface a pu être modelée et mise en forme à l'aide du logiciel "Glade", les fichier ".glade" son les structures des emplacements des boutons et des différents compartiments de la fenêtre.

Elle possède plusieurs compartiments, chacun ayant différentes utilisations. Premièrement nous avons une entête comportant plusieurs volets différents. Lorsque chacun d'entre eux est sélectionné, plusieurs boutons défilent et sont utilisables. Il y a dans un premier tant un volet défilant "fichier" qui va permettre à l'utilisateur de venir sauvegarder son fichier, ou alors venir en créer un nouveau. Un second volet comporte les différents filtre disponible applicable a l'image. Le dernier volet va permettre à l'utilisateur de visiter notre site en lui ouvrant une page web menant à l'URL de notre site. Ensuite nous pouvons retrouver a gauche de la fenêtre, le logo de l'application puis des boutons permettant un retour ou avant suite aux changements effectuer sur la surface. Nous avons par la suite un bouton permettant de changer la couleur des outils. Chaque outils possèdent son propre bouton avec une images simple qui la représente comme par exemple le stylo, la gomme, le seau ou la pipette . En dessous, de ces outils, nous avons une barre glissante permettant a l'utilisateur de modifier l'épaisseur de la ligne de l'outil utilisé par exemple l'épaisseur du stylo ou de la forme. Des boutons avec les différents formes disponibles sont aussi présent par exemple le carré, le triangle ou l'étoile. Nous avons aussi mit a disposition une forme qui s'appelle polygone qui prend en valeur le numéro de coté inscrit par l'utilisateur dans la barre de texte et créera donc une forme avec ce nombre. Enfin, des boutons permettant de changer de formats de la feuille par exemple feuille blanche, quadrillé ou des partitions de musique sont présents tout en bas a gauche de l'application.

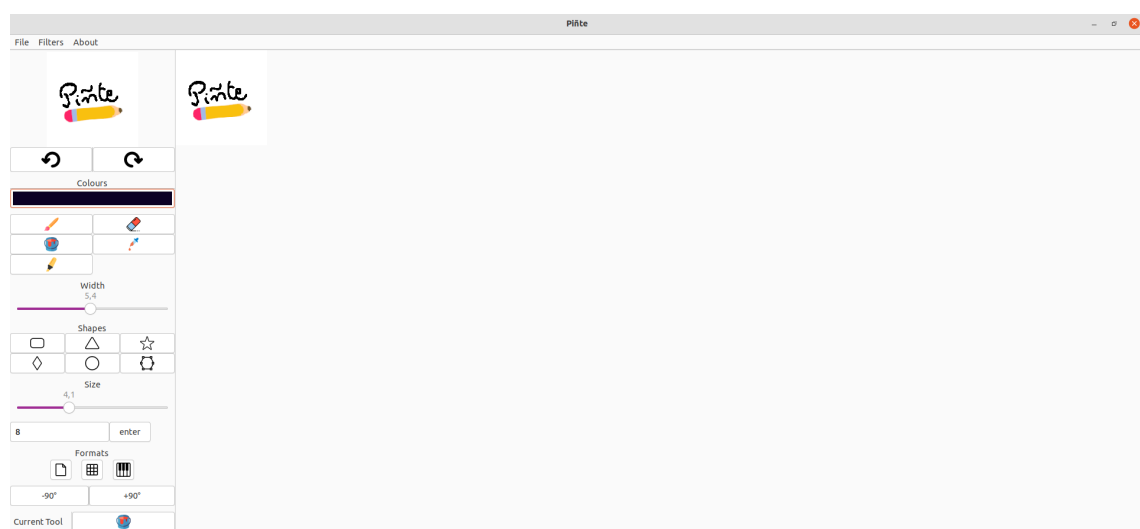


FIGURE 9 – Interface

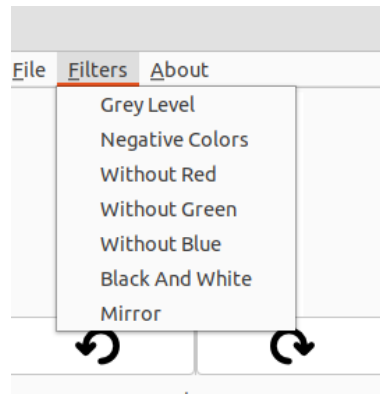


FIGURE 10 – Filtres

4.2 Les outils

Crayon

L'outil crayon servira à l'utilisateur de se mettre en mode dessin. Lorsqu'il appuiera sur le clique gauche de sa souris il pourra faire des traits de la couleurs de son choix, sélectionnable à partir de la palette de couleurs. Cet outil a été mit en place via la détection du clique de la souris et de sa position exacte dans la fenêtre. Une fonction de type "gboolean" "on_click" va venir se connecte via un retour de signal "G_CALLBACK_". Cette fonction va localiser l'emplacement de la souris pour ensuite mettre le point de départ de Cairo sur cette emplacement afin de l'utiliser comme point de départ et par la suite pouvoir faire une ligne entre le point de départ et le point d'arriver grâce a Cairo. Lors du clique et du déplacement de la souris, une ligne se tracera de la première position du clique a sa nouvelle coordonnée lors du maintien du clique , ce qui permettra de reproduire plusieurs petites lignes afin de représenter le dessin de l'utilisateur.

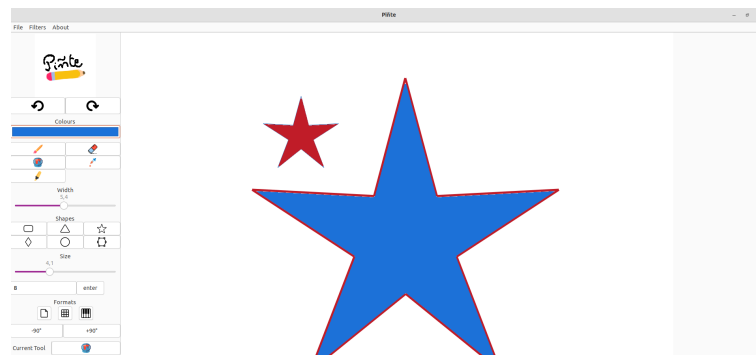


FIGURE 11 – Mode Dessin

Surligneur

Un deuxième stylo est mis à disposition, il s'agit du surligneur. Cet outil fonctionne de la même manière qu'un stylo mais pour donner l'aspect transparent à la couleur, il a fallu modifier la composante alpha d'un pixel de type GdkRGBA.

La première question à se poser est quel niveau de transparence choisiront-nous ?



FIGURE 12 – Niveaux de transparence

Comme vous pouvez le constater sur les rectangles ci-dessus, le niveau de transparence n'est pas le même partout. La transparence d'une couleur possède plusieurs niveaux. Le premier niveau est celui où l'utilisateur aura parcouru un coup de stylo. Le deuxième niveau sera donc celui où l'utilisateur aura parcouru un deuxième coup de stylo sur une couleur précédemment coloriée.

Ainsi, plus le premier niveau sera transparent, plus il sera possible de repasser par-dessus jusqu'à atteindre une composante alpha à 1.0, donc une transparence nulle. Pour cela, nous avons choisis d'initier la composante alpha de la couleur à 0.5 pour le surligneur. Inférieur à celle-ci, le stylo ne serait quasiment pas visible sur la zone de dessin, et supérieur à celle-ci, le stylo serait trop similaire au stylo de base pour écrire.

Ensuite, il a fallu détecter si l'utilisateur coloriait sur une surface non coloriée ou s'il coloriait sur une surface possédant déjà un ou plusieurs niveaux de transparence. Pour ce faire, il a fallu récupérer le pixel courant et voir si les composantes RGB étaient égales. Si oui, il y avait déjà un niveau de transparence sur la surface, et donc il fallait augmenter la composante alpha de la couleur pour pouvoir augmenter l'opacité de la couleur.

En conclusion, nous obtenons un stylo surligneur réaliste, c'est-à-dire que la couleur

s'empile avec celle déjà présente sur la surface.

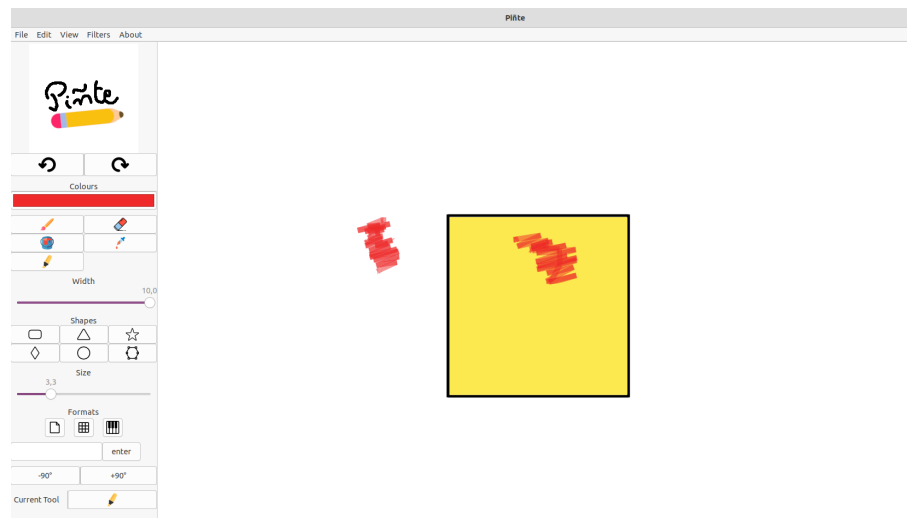


FIGURE 13 – Mode surligneur

Palette de couleurs

Une palette de couleurs est présente, permettant à l'utilisateur d'avoir un large choix de couleur à sa disposition mais aussi de pouvoir choisir une couleur à partir des valeurs de rouge, vert et bleu, lui permettant ainsi d'avoir accès à une infinité de couleurs.

La palette de couleur est un "ColorButtonChooser", le bouton est connecté via un signal et lorsque l'utilisateur va venir cliquer dessus, une fenêtre s'ouvrira. Une structure de type "rgba" nommée "color" a été créée au préalable afin de stocker les différentes caractéristiques de la couleur actuelle.

Cette variable est une structure et elle possède quatre variables, trois pour les couleurs rouge, vert, bleu, un canal "alpha" pour déterminer la transparence de la couleur. Après avoir sa couleur, l'utilisateur va pouvoir dessiner avec la couleur souhaitée, elle sera sauvegardée dans les quatre variables citées.

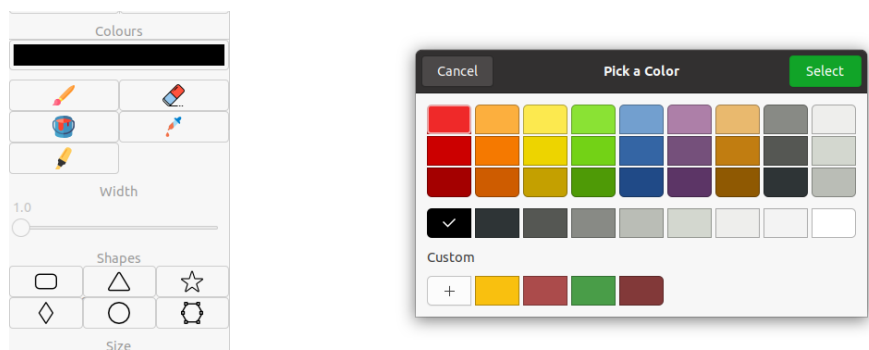


FIGURE 14 – palette de couleurs

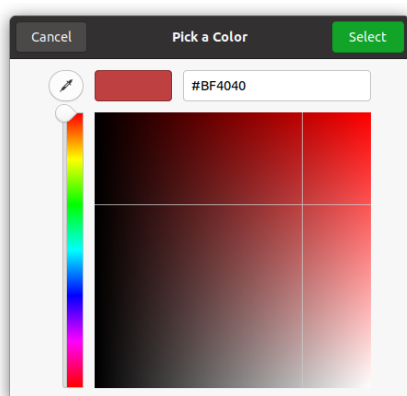


FIGURE 15 – palette de couleurs

Pipette

L'outil pipette est un outil très utile et simple d'utilisation. En effet il s'agit de récupérer une couleur présente sur la surface de dessin. L'utilisateur sélectionnera l'outil, puis il sera libre de choisir une couleur parmi tous les pixels présents sur la zone de dessin, il appuiera ensuite sur le clic gauche de la souris, pour confirmer son choix et la couleur se mettra directement en couleur choisi pour les outils.

L'outil pipette sert à récupérer une couleur déjà présente sur la zone de dessin. Le principe est très simple : il suffit de récupérer le pixel en position (x,y) au moment du clic. En récupérant le pixel, nous avons accès au trois composantes RGB qui nous intéressent. Dans notre code, nous avons mis trois variables en publics : red, green et blue. Ces trois variables nous indiquent la couleur actuelle que l'utilisateur a choisi. Ainsi, lorsque nous récupérons la couleur du pixel au clic, il suffit de changer les valeurs des trois variables en celles des composantes RGB du pixel. De cette manière, nous avons modifier la couleur courante en celle que nous voulons obtenir. Le problème restant est de également modifié la couleur affichée sur la palette. Pour récupérer une couleur sur Gtk, nous avons à disposition la fonction « `gtk_color_chooser_get_rgba` », pour attribuer une couleur sur Gtk, nous avons à disposition la fonction « `gtk_color_chooser_set_rgba` ». Nous utiliserons donc la deuxième fonction qui modifiera l'affichage de la palette.

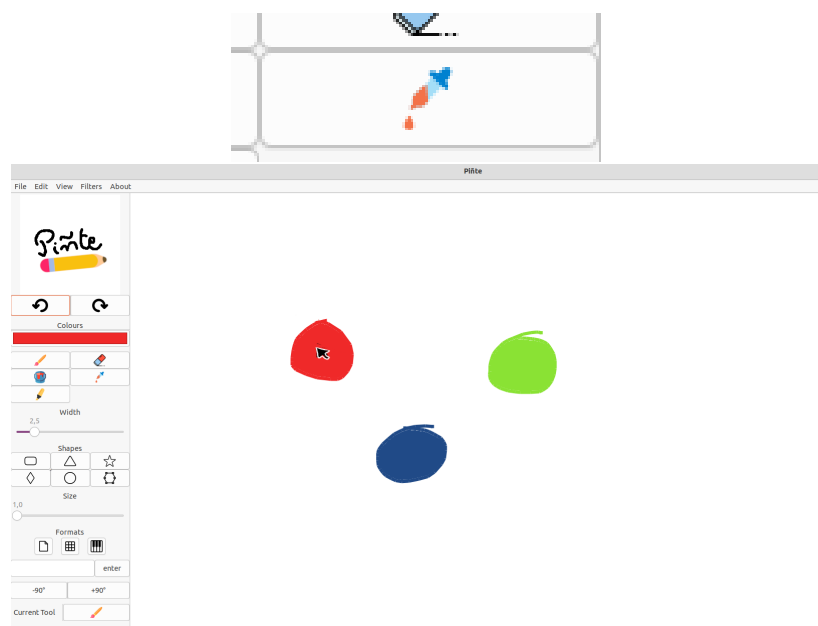


FIGURE 16 – pipette

4.2.1 Formes

Les formes sont des outils pratiques afin de créer de la diversité sur le dessin de l'utilisateur. Nous avons une panoplie d'objet partant d'un simple jusqu'à un polygone personnalisable. Chaque formes pourrait être dessinée en ayant choisie au préalable sa taille mais aussi l'épaisseur de ses contours.

The image shows a user interface for drawing shapes. It consists of several sections:

- width**: A slider control with a circular knob at the left end. The value **1,0** is displayed to the left of the knob.
- Shapes**: A grid of six shape icons arranged in two rows and three columns:
 - Row 1: A rounded rectangle, an equilateral triangle, and a five-pointed star.
 - Row 2: A diamond (rhombus), a circle, and a hexagon with small squares at its vertices, indicating it is a customizable polygon.
- Size**: A slider control, identical in design to the width slider, with the value **1,0** displayed to its left.
- Input Field**: A numeric input field containing the number **8**, followed by an **enter** button.

FIGURE 17 – Formes

Carre

Le carre est créer en récupérant les coordonnées de la souris. Créant un point vers la droite avec une taille "size", de même vers le haut. Et ensuite avec les coordonnées additionnées avec "size", partant de ce point, les relier au point expliqués précédemment.

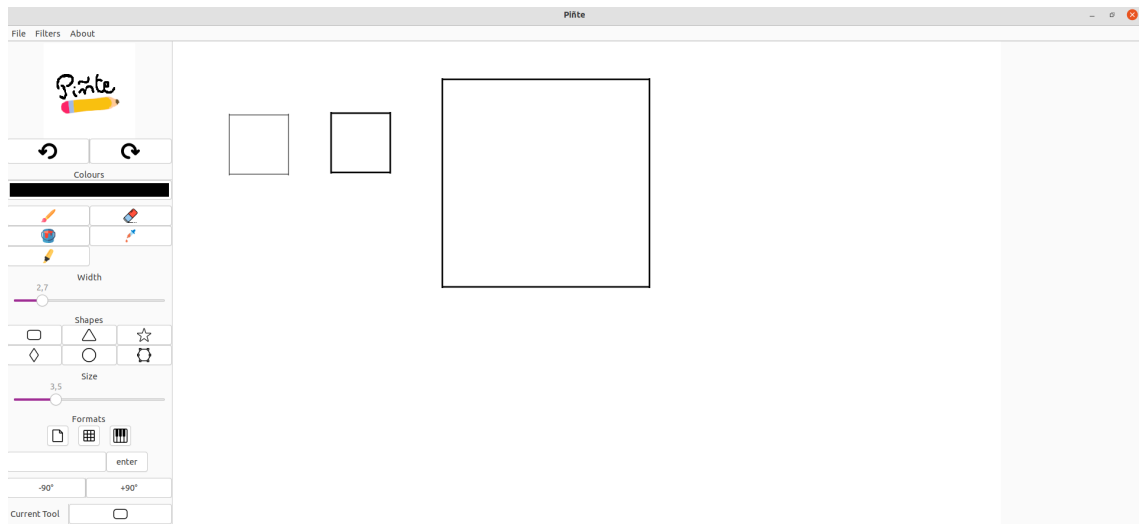


FIGURE 18 – Carré

Triangle

Afin de créer un triangle équilatéral il a fallu compiler trois différents points. Le premier se situe au coordonnées de la souris. Le deuxième est calculer en fonction de la position de la souris et décaler d'une taille "size" sur l'axe des abscisses de la fenêtre. ayant pour valeur 100, notre variable "size" est modifiable a l'aide d'un multiplicateur sur l'interface utilisateur, qui peut prendre un amplitude de 1 à 10. Ensuite le troisième point prend donc la moitié de cette taille "size" pour l'axe des abscisses. Pour sa position sur l'axe des ordonnées, a l'aide du théorème de Pythagore, le point sur l'axe des ordonnées est : $\text{"mousey"} - \sqrt{(\text{size} * \text{size}) - (\text{size} * \text{size})/4}$ ", avec "mousey" la position sur l'axe des ordonnées de la souris, "size" la taille expliquée précédemment, et "sqrt" la fonction racine carre de la classe "math.h".

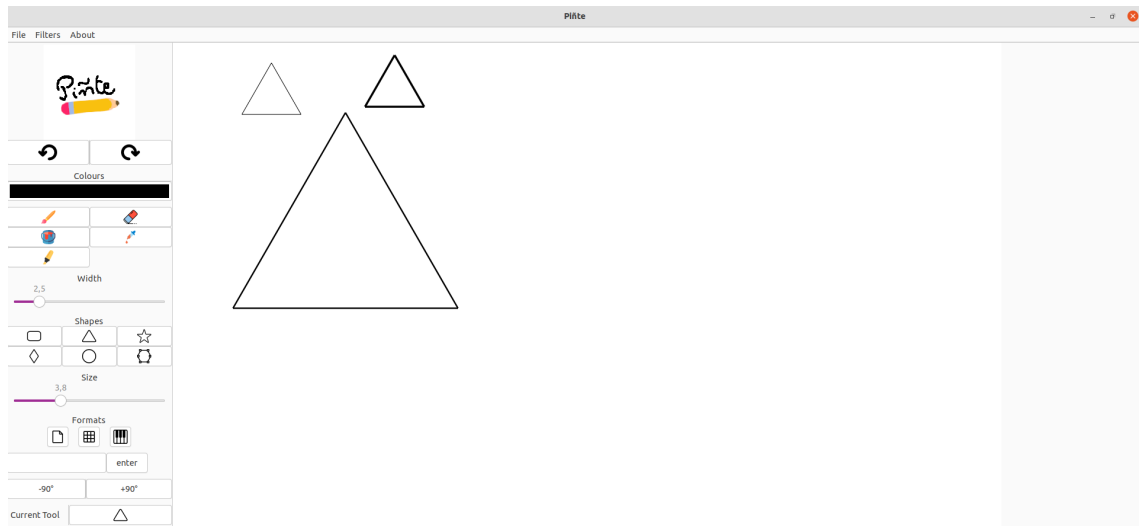


FIGURE 19 – Triangle

Cercle

Avec la méthode `cairo_translate()`, nous déplaçons l'origine du dessin au centre de la fenêtre. Nous voulons que notre cercle soit centré. Pour dessiner le cercle nous utilisons la fonction `cairo_arc()` déjà présent dans la bibliothèque permettant de dessiner des arc, il nous suffit de dessiner un arc avec une courbe de $2 \times \pi$.

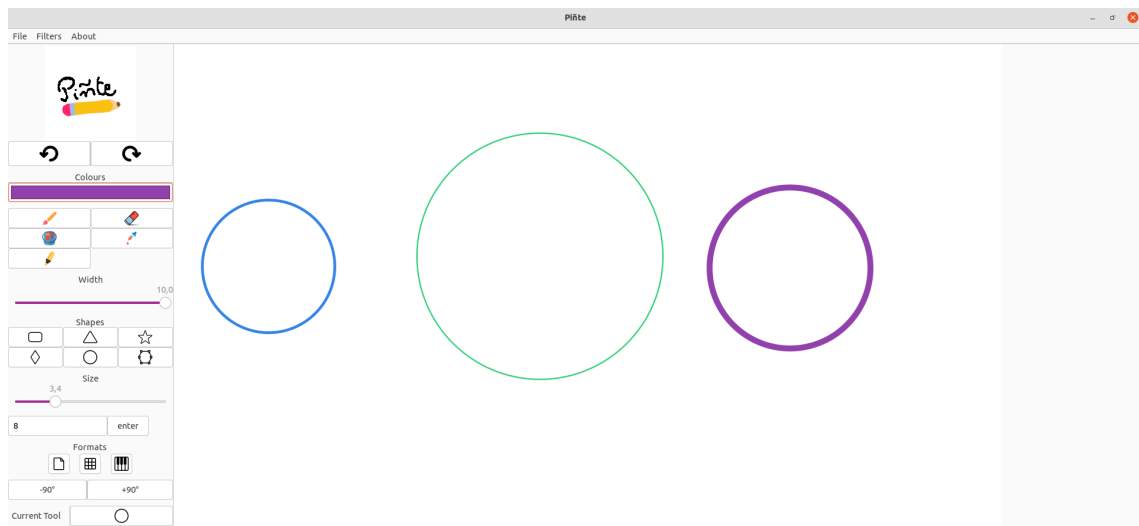


FIGURE 20 – Cercle

Polygone

Le polygone est une forme aux possibilités infinies. En effet, on peut choisir le nombre de côtés que le polygone va avoir, et en utilisant la formule $(r * \cos(2 * \pi * k/n + \pi/2))$ pour la coordonnée horizontale et $(r * \sin(2 * \pi * k/n + \pi/2))$ pour la coordonnée verticale, avec r le rayon du polygone, k le numéro du côté du polygone et n le nombre de côtés du polygone. Les fonctions cosinus et sinus ainsi que ont été importes depuis ma bibliothèque `<math.h>`. On y ajoute les coordonnées x et y de la souris et un polygone va se dessiner. Les différents sommets calculés seront stockés dans deux listes, une pour les coordonnées horizontales et l'autre pour les coordonnées verticales. Les sommets sont enfin reliés l'un après l'autre.

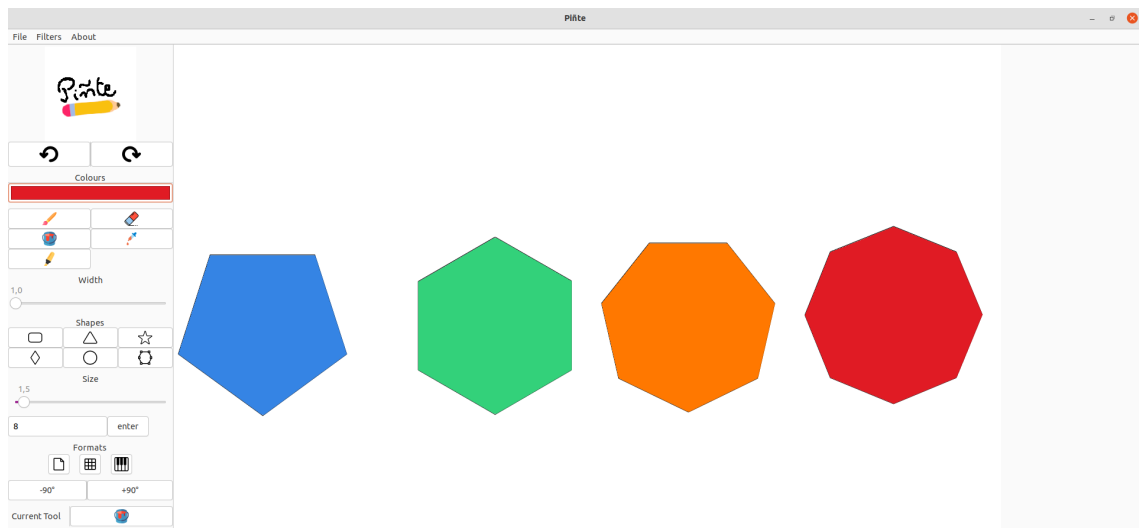


FIGURE 21 – Polygone

Étoile

Notre forme étoile est également un polygone de 5 côtés, avec cependant la particularité d'avoir en plus 5 autres points plus petits. Le calcul des coordonnées de ces points ne diffèrent pas beaucoup du calcul des extrémités : $(r/3 * \cos(2 * \pi * k/5 + \pi/2 + 2\pi/10))$ pour la coordonnée horizontale et $(r/3 * \sin(2 * \pi * k/5 + \pi/2 + 2\pi/10))$ pour la coordonnée verticale, avec r le rayon du polygone et k le numéro du côté du polygone. Les fonctions cosinus et sinus ainsi que ont été importes depuis ma bibliothèque `<math.h>`. Le rayon est divisé par 3 pour former une étoile. On utilisera ici quatre listes pour les coordonnées des extrémités et des sommets intérieurs, afin de plus facilement les relier. La possibilité de faire une infinité d'étoile était également possible, mais on n'a pas trouve l'idée assez pertinente.

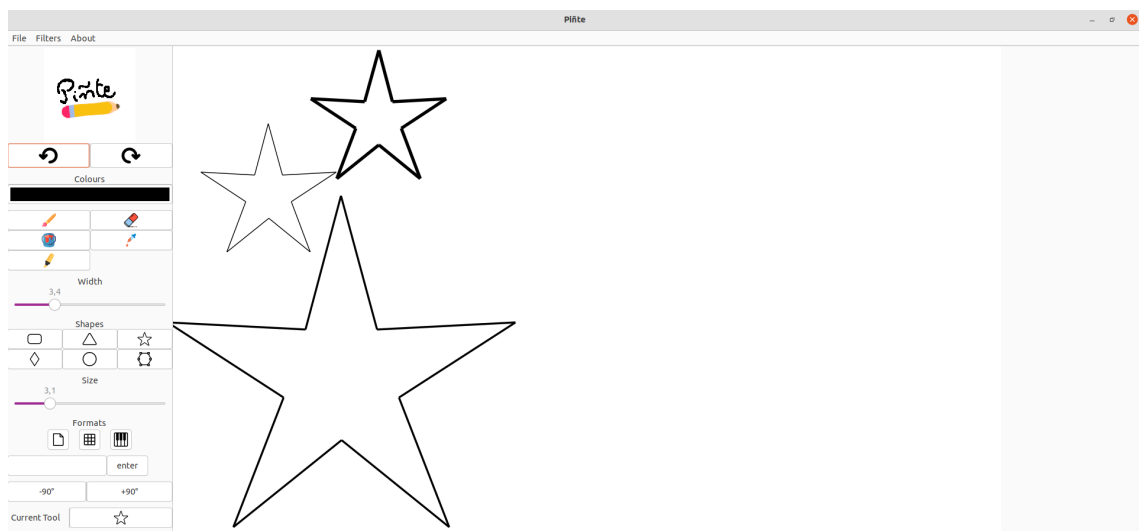


FIGURE 22 – Étoile

Losange

La méthode pour dessiner un losange va etre similaire a celle du triangle, nous partons du point ou se situe la souris. Puis nous compilons un point espace de "size" vers les X positif sur l'axe des abscisses. Puis nous créons un point de distance "size/2" sur l'axe des abscisses et de même longueurs sur l'axe des ordonnées. Nous n'avons plus qu'a relier les sommets entres eux.

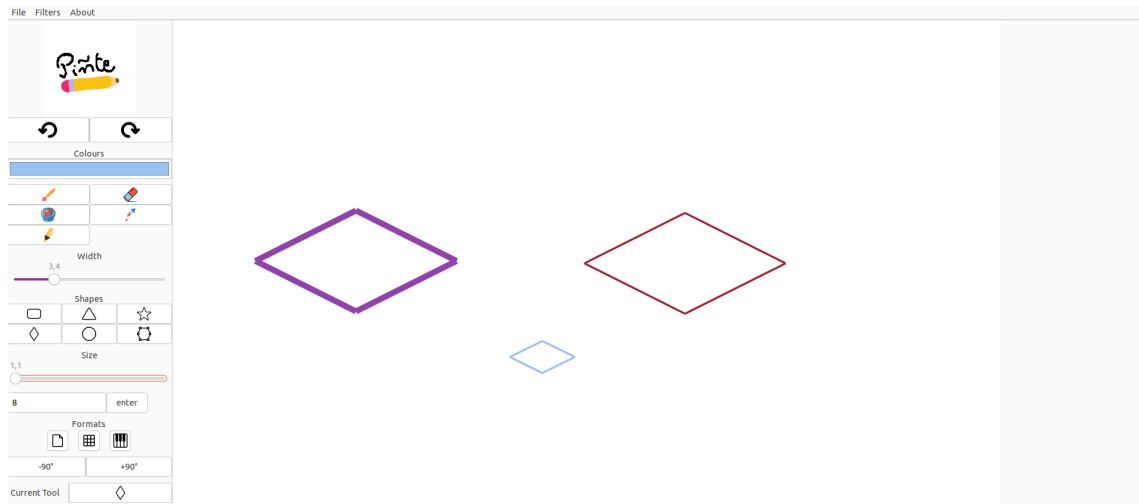


FIGURE 23 – Interface

4.2.2 Formats de la page

Pour nous différencier des autres applications de dessins, nous avons décidé de rajouter plusieurs formats possibles pour l'utilisateur : le format page blanche, le format page quadrillée et le format page de musique. Nous avons la possibilité d'en rajouter un nombre incalculable mais pour le moment, seul ces trois formats sont à disposition.

La diversité des formats à pour but de varier l'utilité de notre application, nous voulions qu'un enfant puisse utiliser l'application pour dessiner, tout comme un musicien peut utiliser notre application pour faire des partitions de musique.

Pour changer de surface, nous avons dans notre répertoire plusieurs images aux mêmes dimensions qui représentent chaque surface. Ainsi, lorsque l'utilisateur clique sur le bouton « quadrillage », le programme va ouvrir et afficher l'image associé au quadrillage. L'image viendra donc s'afficher sur la zone de dessin ce qui nous les résultats suivants.

Page blanche vierge

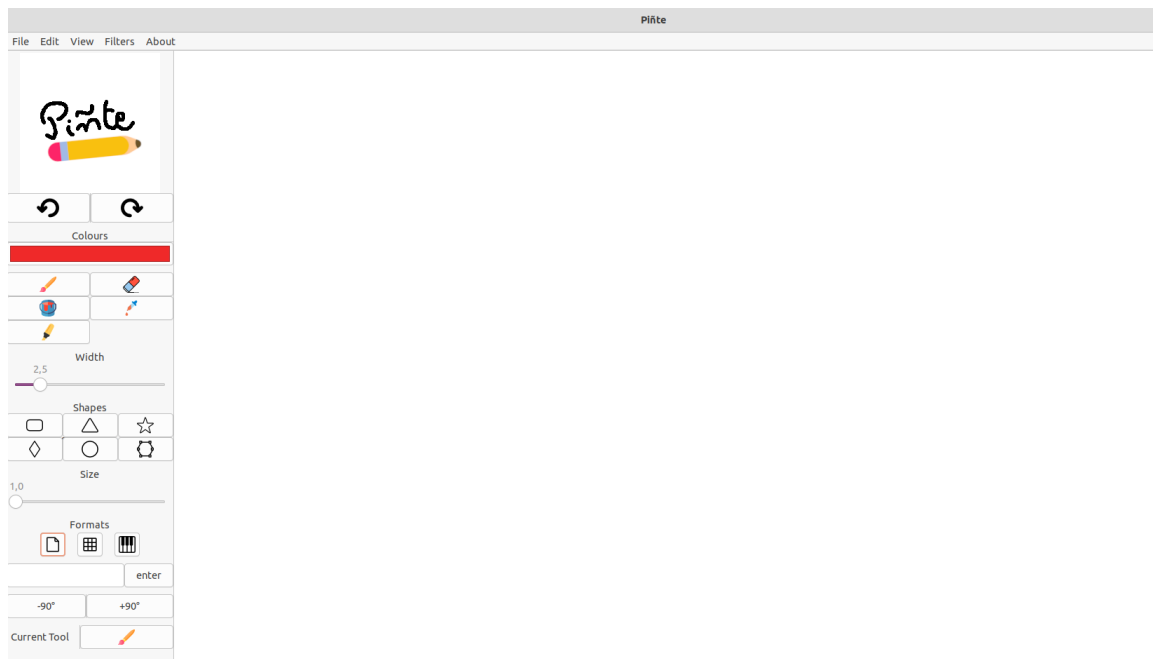


FIGURE 24 – Format page blanche

Page quadrillée

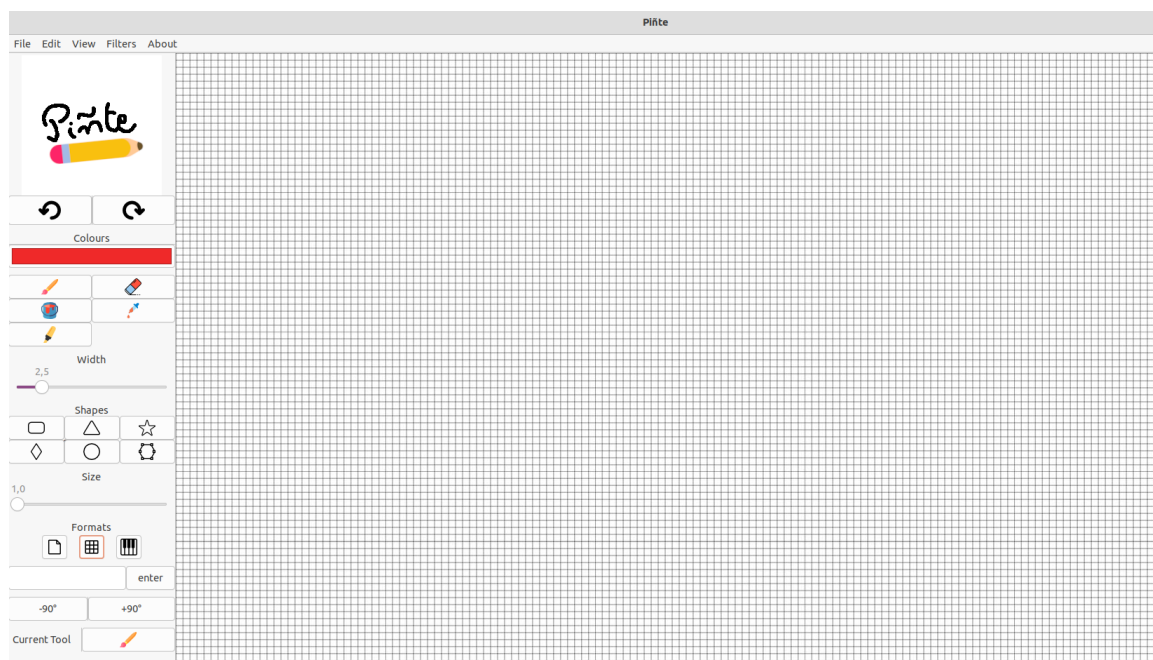


FIGURE 25 – Format page quadrillée

Page de musique



FIGURE 26 – Format page de musique

Le seau

Le seau permettra à l'utilisateur de pouvoir remplir une surface avec une couleur définie. Cet outil est indispensable car il permettra un coloriage rapide et précis. En effet, il sera plus efficace d'utiliser cet outil que d'utiliser le crayon pour colorier manuellement chaque pixel d'une surface.

Nous utiliserons l'algorithme de remplissage par inondation. Il est utile dans les cas où il n'y a pas de limite de couleur unique pour le polygone, c'est-à-dire que la surface a plusieurs couleurs. Dans l'algorithme de remplissage par inondation, au lieu de remplir une couleur de délimitation spécifique, nous remplissons simplement les pixels avec la couleur par défaut. Il s'agit de l'algorithme utilisé dans l'outil de remplissage "bucket" des programmes de peinture comme Paint pour remplir des zones connectées, de couleur similaire, avec une couleur différente.

```

1 # FloodFill algorithm
2 FloodFill(pixel, target-color, replacement-color)
3     if target-color == replacement-color
4         return
5     if target-color != pixel->color
6         return
7
8     pixel->color = replacement-color
9
10    FloodFill(pixel->y+1, target-color, replacement-color)
11    FloodFill(pixel->y-1, target-color, replacement-color)
12    FloodFill(pixel->x+1, target-color, replacement-color)
13    FloodFill(pixel->x-1, target-color, replacement-color)
14
15    return

```

FIGURE 27 – Pseudocode de l’algorithme de remplissage par inondation

L’algorithme de remplissage par inondation a plusieurs méthodes d’implémentation :

1. 4-connected pixel
2. 8-connected pixel

4-connected pixel

Dans la méthode du 4-connected pixel, nous devons vérifier les 4 pixels adjacents au pixel actuel, nous appelons ces quatre directions : nord (haut), sud (bas), ouest (gauche) et est (droite). On remplit les pixels suivant les cinq étapes ci-dessous :

1. Initialiser les quatre valeurs X, Y, Fill-color et Default-color. X et Y sont les coordonnées de la position de la souris au clic, Fill-color représente la couleur dont nous voulons remplir les pixels, Default-color représente la couleur par défaut : le blanc sur une page vierge, le noir sur une page avec un fond noir, etc...
2. Si la couleur du pixel n’est pas égale à Default-color, nous quittons la fonction.
3. Associer Fill-color au pixel courant.
4. Exécuter récursivement la fonction pour chaque pixel adjacent, donc vers le sud, le nord, l’ouest et l’est du pixel.
5. Quitter la fonction.

Nous avons utilisé cet algorithme qui est pour le moment suffisant pour être présenter. Mais l’algorithme 4-connected pixel possède quelques défauts d’utilisation.

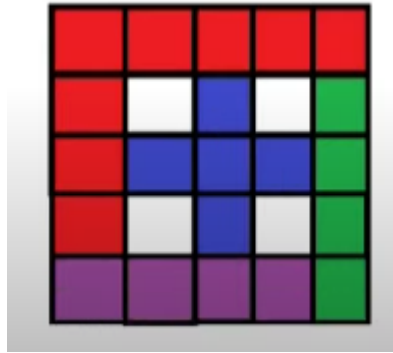


FIGURE 28 – 4-connected pixel

En effet, en parcourant seulement le nord, le sud, l'ouest et l'est du pixel, nous allons vers quatre pixels adjacents. Ce qui peut laisser les pixels non coloriés sur les bordures des formes à colorier. Pour remplir correctement les pixels, il faut également vérifier les pixels sur le coin gauche supérieur, le coin droit supérieur, le coin gauche inférieur et le coin droit inférieur.

C'est de cette manière que le deuxième algorithme 8-connected pixel sera plus performant.

8-connected pixel

Comme indiqué dans son nom, cette méthode étudiera les 8 pixels adjacents à un pixel. Bien évidemment, cet algorithme sera plus long dans l'exécution car plus de pixels seront vérifiés mais son résultat sera parfait.

Pour vérifier les quatre coins du pixel courant, nous manipulerons les pixels en position :

1. $(X+1, Y+1)$: coin inférieur droit
2. $(X+1, Y-1)$: coin supérieur droit
3. $(X-1, Y+1)$: coin inférieur gauche
4. $(X-1, Y-1)$: coin supérieur gauche

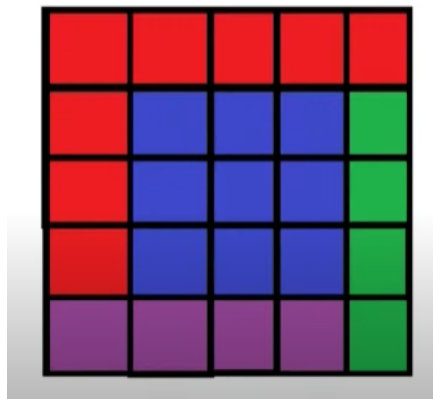


FIGURE 29 – 8-connected pixel

Notre outil *bucket* nous permet donc d'effectuer toutes les actions dont un utilisateur s'attendrait à avoir sur une application de dessin.



FIGURE 30 – Démonstration de l'outil *seau*

L'outil est fonctionnel mais nous corrigerons les actuels problèmes d'optimisation pour améliorer l'ergonomie de notre application.

Taille du crayon

Afin que l'utilisateur puisse modifier la taille de son crayon à sa guise et sans difficultés, un outils très pratique a été mis en place. Le modificateur de taille est un "slider", c'est-à-dire un bouton qui va bouger sur un raille d'une valeur à une autre. Premièrement un "GtkAdjustement" est crée, il correspond à une mesure utilisable sur des objets, comme ici, où nous l'utilisons pour en faire ce slider. Il est défini pour avoir une valeur initiale de 1.0. On définit aussi son amplitude qui va des valeurs 0.0 à 10.0. En complément peut être modifié le pas du changement qui est de 1.0, c'est-à-dire que lorsque l'utilisateur utilisera la flèche droite de son clavier pour modifier la taille, le slider ira de 0.1 en 0.1. Et le grand pas du slider est de 1.0 en 1.0 qui fonctionne pareil que le précédent sauf que l'utilisateur devra utiliser la touche flèche du haut.

Suite a ça, cet Adjustment va être injectée dans un "gtk_scale". Ce widget est initialise avec un "Adjustment" en paramètre, un champ qui va demander si nous voulons l'utiliser en mode horizontal ou en mode vertical. De plus il fallu préciser ou est-ce que nous voulions l'insérer dans notre fenêtre, ici nous avons du utiliser un autre méthode car Glade ne permettait pas d'insérer un "Adjustment" directement. Une dernière instruction est utilisée "gtk_container_add" qui va insérer dans une des cases de l'application notre slider. Nous avons donc renseigner une case spécifique où le slider va être initialiser au lancement de la fenêtre.



FIGURE 31 – Changer la taille du stylo



FIGURE 32 – Exemples de taille

Undo/Redo

Avec l'utilisation d'une Stack, on a pu implémenter un système de retour en arrière et en avant. La Stack va se remplir avant chaque action de l'utilisateur, en s'ajoutant la surface convertie en pixbuf a ce moment précis. Elle aura une taille maximale de 10 pixbuf, chaque action au de-la de cette limite va effacer le pixbuf le plus ancien et ajouter le nouveau pixbuf. Ainsi lorsque le bouton Undo est activé, on va prendre le dernier pixbuf et remplacer la surface actuelle. L'ancienne surface est stockée, au cas ou l'utilisateur veut revenir en avant en activant le bouton Redo. La taille du Stack peut évidemment être modifiée. On s'est inspiré d'un ancien TP sur les queues, on a repris les fonctions de ce TP afin de modifier notre Stack.

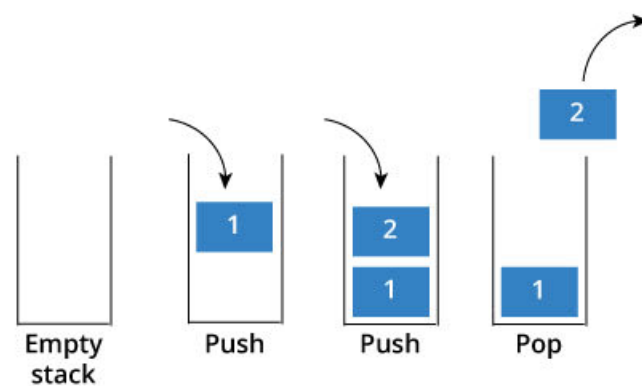


FIGURE 33 – Schéma d'une Stack

Actualisation

Une fonction d'actualisation de zone la de dessin a aussi été implémenté afin d'avoir la bonne surface afficher a chaque modification de surface. Cette fonction met a jour la surface actuelle de la drawing area en ajoutant dans la bonne surface dans la file.

4.2.3 Filtres

Grâce au filtre, l'utilisateur peut changer le visuel de l'image. Par défaut, notre application propose différents filtres comme une image en niveau de gris, une image plus contrastée, une image plus lumineuse, une image avec un niveau de rouge plus important, etc...

Ces filtres sont créés à partir de l'image sur la drawing area, qui va être récupérée et parcourue pour enfin modifier les couleurs des pixels de ce dernier.

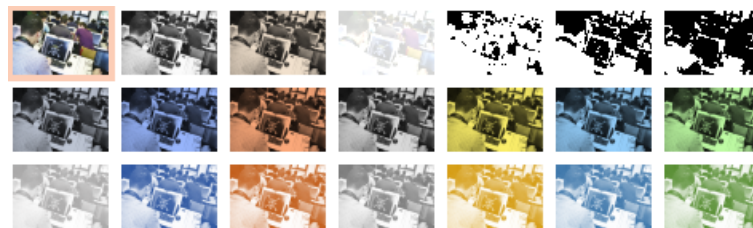


FIGURE 34 – Filtres à disposition

Pour le moment, nous possédons sept filtres :

1. Niveaux de gris



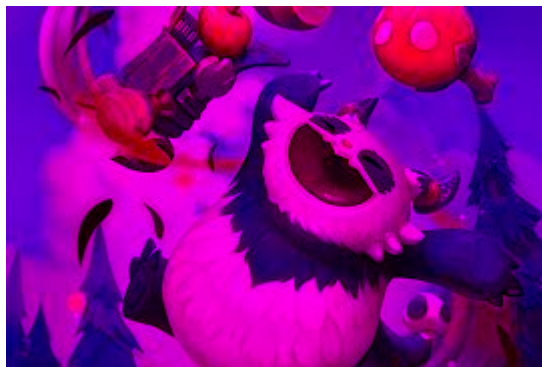
2. Inversion des couleurs



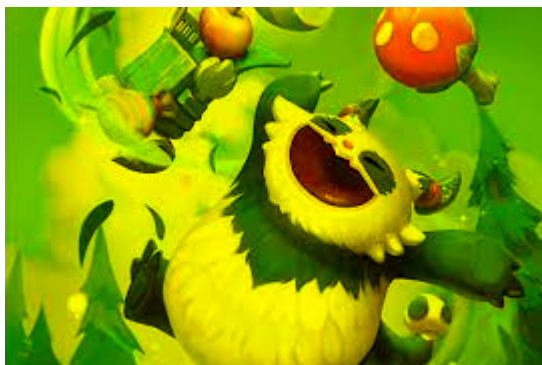
3. Sans composante rouge



4. Sans composante verte



5. Sans composante bleue



6. Noir et Blanc



7. Miroir



Pour avoir un plus large choix, nous comptons récupérer les filtres ci-dessus et modifier leurs contrastes et leurs luminosités.

4.2.4 Conversion de l'image pour les filtres

Afin d'appliquer les filtres sur notre drawingarea actuelle, il a fallu faire des conversion de l'image. Premièrement on a récupéré les pixbuf correspondant à l'image, pour ensuite lui la charger sur une initialisation d'une surface SDL. Suite à ça les différentes fonction de filtres sont appelées sur cette surface. Afin de renvoyer l'image modifier sur notre drawingarea, il faut que l'image passe du format ".bmp" au format ".png". Une fonction de conversion d'image a donc été créée et elle va nous renvoyer notre image ".bmp" sous son nouveau format ".png". Notre image en ".png" est enfin injectée dans le pixbuf.

4.2.5 Rotation

Un bouton rotation $+90^\circ$ et -90° on été ajouter permettant a l'utilisateur de tourner l'image vers la droite ou vers la gauche selon ses envies. La fonction de rotation fonctionne de la même manière que pour les filtres, c'est a dire que nous récupérer l'image affiché sur la drawing area et que nous modifions cette image pour enfin l'afficher de nouveau sur la drawing area. Donc pour appliquer, cette rotation il nous a fallut faire créer une nouvelle surface ayant pour hauteur la largeur et en largeur la hauteur de l'image a tourné. Par la suite, nous avons parcourut l'image a tourne et nous avons récupérer le pixel en "x,y" pour le mettre dans la surface vierge en "hauteur - x - 1, y".



FIGURE 35 – Boutons de rotation

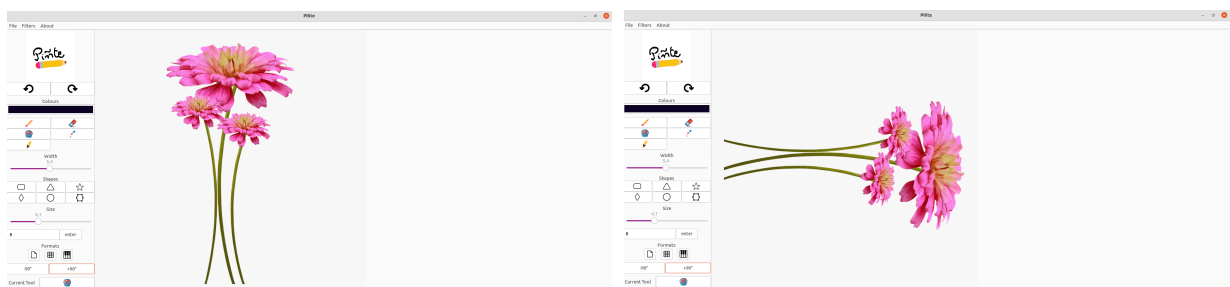


FIGURE 36 – rotation

4.2.6 Système de sauvegarde de fichier

Un système de sauvegarde de fichier à été implémenté permettant à l'utilisateur de sauvegarder son fichier lorsqu'il le souhaite. En convertissant notre surface en pixbuf, on peut également choisir l'emplacement où l'image va être stockée et choisir son nom, avec l'aide de GTK. La sauvegarde de l'image, quant à elle passe par une fonction de manipulation de pixbuf.

4.2.7 Système de sauvegarde automatique

Un système de sauvegarde automatique a été mis en place pour pouvoir récupérer la dernière surface modifier. Cela permet a l'utilisateur de ne pas perdre son travail s'il ferme malencontreusement l'application ou que son ordinateur s'éteint. L'image de la sauvegarde automatique se trouvera dans les fichiers de notre application, et qui sera accessible par l'utilisateur.

4.2.8 Outils en cours d'utilisation

Nous avons remarquer qu'il était facile de se perdre lorsque l'on utilise l'application, notamment l'outil que nous sommes actuellement en train d'utiliser. Pour remédier à ça, nous avons décider d'implémenter en bas a gauche de l'écran un petit icône qui va montrer a l'utilisateur l'outil qu'il est en train d'utiliser. Ce bouton fonctionne se met a jour facilement lorsque l'utilisateur va sélectionner un des outils au dessus. C'est donc un "GtkButton" avec une personnalisation spéciale, qui va être renseigner à chaque sélection d'outil. Le même icône de l'outil sera donc affiché.

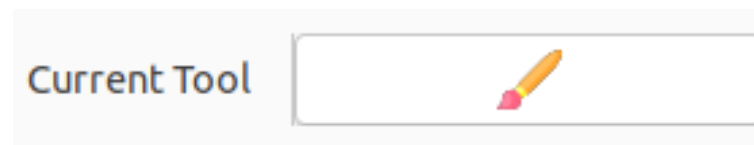


FIGURE 37 – Outil en cours d'utilisation

4.3 Lancement de l'application

Chargement de l'image

Si l'utilisateur veut éditer une image, il doit charger une image au début de son travail. Pour charger une image, il y aura un bouton permettant de choisir une image depuis son ordinateur. L'image pourra ainsi être modifiée depuis notre application. Pour pouvoir charger une image, nous utilisons la fonction de GTK `gtk_file_chooser_get_filename` afin de récupérer l'emplacement du fichier sélectionné pour ensuite l'utiliser dans fonction de chargement de l'image qui modifiera la zone de dessin en cette dernière. Cependant lors du chargement de l'image nous avons du aussi modifier la longueur et largeur de notre fenêtre de dessin afin que cette fenêtre soit de la même taille que l'image.

Édition

Lorsque l'utilisateur aura chargé son image a partir de ses fichiers, il la verra sur notre surface de départ, prête a subir des modifications. En effet, on pourra dessiner par dessus.

Pour éditer l'image, nous avons tout d'abord transformé l'image en un tableau de pixels pour pouvoir modifier ce tableau nous l'avons converti en surface afin de pouvoir modifier cette surface grâce à la fonction de dessin.

5 Site internet

Le site Web de notre projet est avant tout le site Web de notre groupe (ALPH) et aussi de notre application. Ceux-ci sont respectivement représentés au travers de différents articles ainsi que liens de téléchargements et autres interactions dans le but étant que les visiteurs aient envie de visiter le site et donc de télécharger l'application.

La structure du site Web est entièrement terminé mais sera mis à jour suite aux nouveaux ajouts et est disponible à l'adresse suivante :

https://akaagi.github.io/Pinte_Website/accueil.html

5.1 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 ainsi qu'un bouton « Télécharger » (Accueil, Pinte, Documentation, À Propos et Contact), puis dans la suite de la structure un corps de texte qui laisse libre recours à nos choix et enfin un pied de page destiné à n'être qu'un simple décor.

5.1.1 La page d'accueil :

Bien en évidence, nous avons opté pour directement placer un bouton afin de renvoyer vers un lien de téléchargement de l'application. Dans le corps de la page, nous avons opté pour une présentation brève de notre groupe ALPH répartie en différents onglets mis en parallèle avec des images représentatives avec une présentation du Studio, puis de nos valeurs en tant que créateur, un petit paragraphe sur notre politique ainsi que le choix de notre Logo.

Plus bas, il nous a semblé important 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 aussi renseigné sous chaque personne.

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

5.1.2 La page de présentation de l'application

En tête de page, nous présentons tout d'abord le logo Pinte créé par nos soins avec directement un bouton pour renvoyer vers le téléchargement de l'application.

Dans le corps de la page nous avons opté pour une présentation de notre application avec notamment une mise en avant de ces qualités telles qu'une prise en main simple, une bonne expérience d'édition et design épuré.

En pied de page nous avons simplement fait le choix de placer le logo de notre équipe.

5.1.3 La page des documents :

Ici nous avons rassemblé l'ensemble des documents qui nous ont été demandés durant l'avancement de notre projet à savoir notre cahier des charges, celui de soutenance une, deux et finale. Elles sont présentées avec leurs différentes caractéristiques avec un bouton pour les télécharger (pour l'instant seul le cahier des charges et de première soutenance est disponible).

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

En pied de page nous avons simplement fait le choix de placer le logo de notre équipe.

5.1.4 La page à propos :

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

5.1.5 La page contact :

Sur cette dernière page, nous avons simplement disposé quatre moyen pour pouvoir nous contacter ou qui concernent la partie communication de notre projet.

5.2 La réalisation du site

Nous avons opté pour choisir un hébergeur gratuit se renseigner sur la façon de développer notre site. 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.

5.3 L'apparence du site

Nous nous sommes basés sur le thème de notre logo pour l'apparence du site qui allie 2 couleurs douces, discrètes mais aussi à la fois efficace et moderne. Le langage CSS n'étant pas toujours évident, cela est 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 la totalité des 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.



FIGURE 38 – Page principale



EDITEUR D'IMAGES

TELECHARGER GRATUITEMENT POUR LINUX

Découvrez Pinte !

Pinte est 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 a une fonction qui permet de charger une image à partir d'un explorateur de fichier et de venir dessiner sur cette dernière ou de la manipuler. Par la suite, cette image pourra être sauvegardée dans les fichiers locaux de l'ordinateur. Nous allons opter pour une interface sobre et simple 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. Le principe de l'application est de mettre l'utilisateur directement sur une surface blanche, sur laquelle il pourra dessiner avec les différents outils présents, mais il pourra également éditer une image qu'il chargera au préalable.

MANUEL D'EXPLOITATION DISPONIBLE ICI

✓

PRISE EN MAIN SIMPLE.
Interfaces pratiques.
Accessibilité à un large publique.

✓

BONNE EXPÉRIENCE D'EDITION.
Nombreux outils graphiques.
Simplicité et efficacité des outils.

✓

DESIGN ÉPURÉ.
Interface sobre.
Simple mais efficace.



FIGURE 39 – Présentation de Pinte

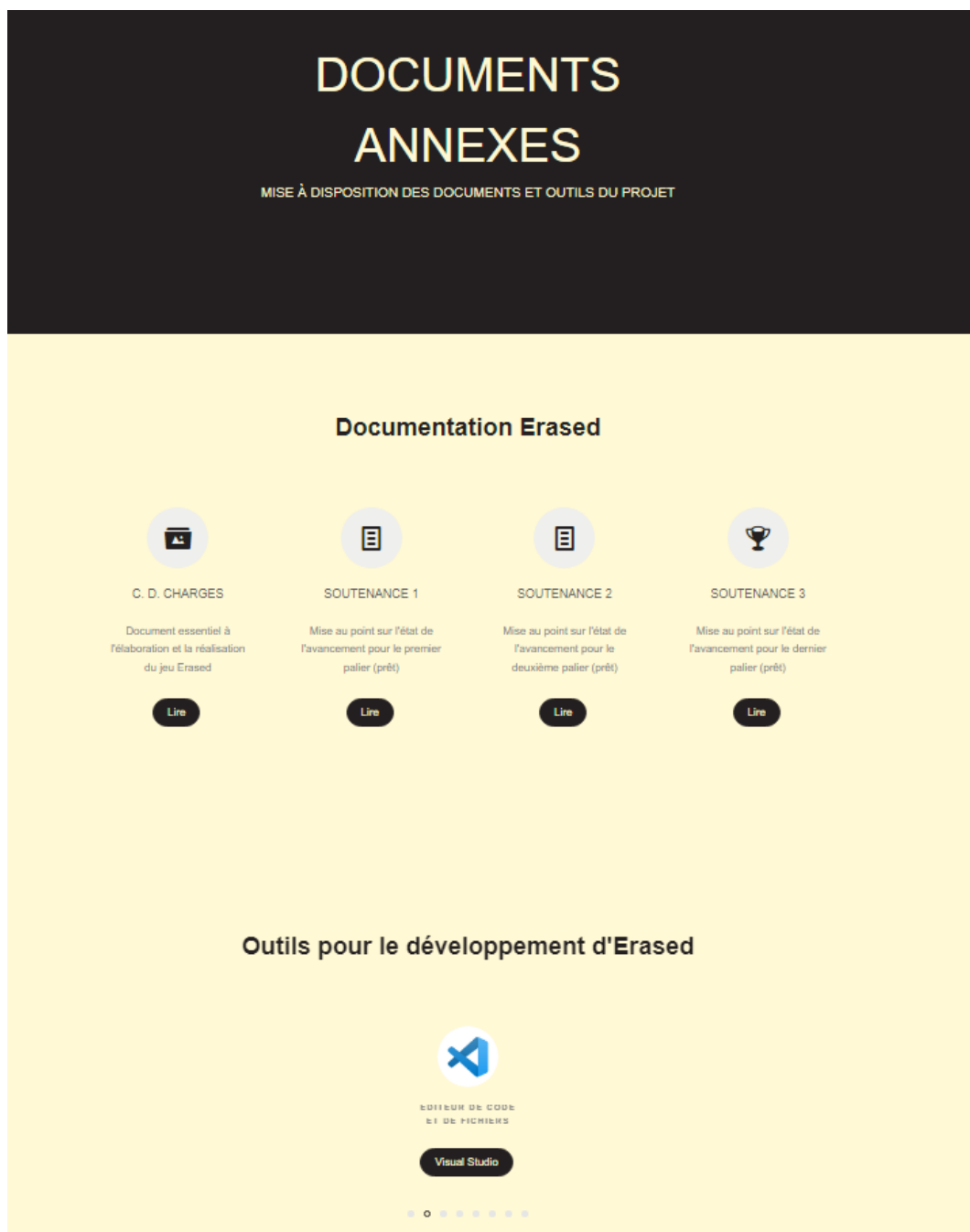


FIGURE 40 – Documentations

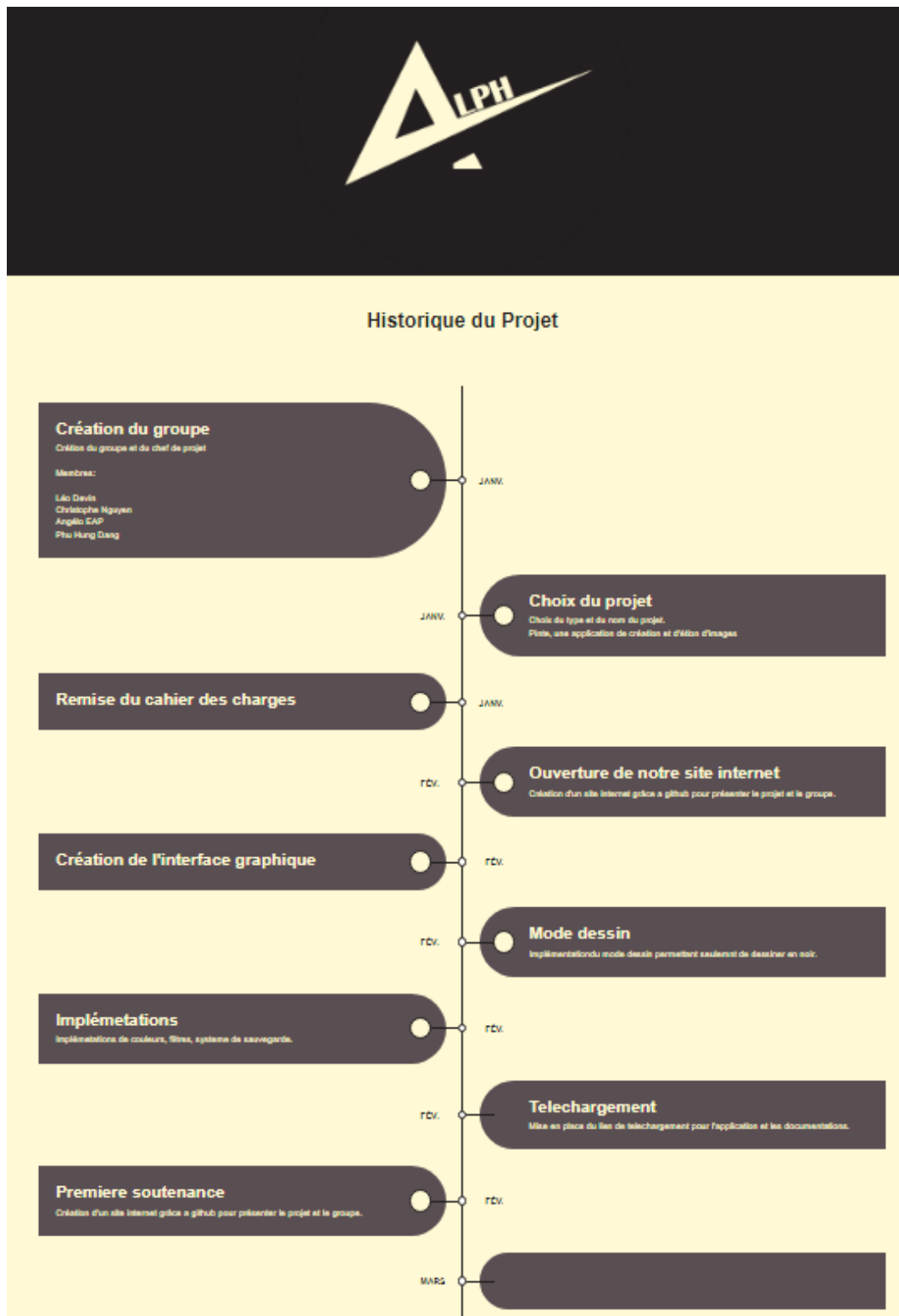


FIGURE 41 – Historique du projet

6 Les différentes tâches

Notre travail s'effectue autour des horaires et des disponibilités de chacun, on essaie de travailler en même temps afin de pouvoir s'entraider de manière efficace et assidu. Un des membres n'étant pas dans la même classe, il est difficile pour nous de pouvoir tous travailler en même temps. En groupe, nous avons tout d'abord choisit le thème de notre application et principalement travaillé sur l'apparence de notre fenêtre. De plus, nous nous sommes aussi réunis pour parler des fonctionnalités que l'on voulait implémenter, ainsi que sur les différents problèmes rencontrés lors des tâches personnelle.

6.1 Travail personnel et ressenti du projet

6.1.1 Léo DEVIN

Au cours de ces cinq mois de travail, mon rôle en tant que chef de projet a été d'organiser et motiver les troupes du au relâchement après avoir fait eu un semestre intense. Pour ce qui est du travail réalisé, j'ai pu premièrement remettre en forme notre fenêtre qui n'était pas si bien organiser jusque la. A l'aide d'Angelo j'ai pu implémenter les premiere formes, le carre, le triangle et losange. Ensuite j'ai eu l'idée d'implémenter un indicateur permettant a l'utilisateur de savoir quel outils il est en train d'utiliser. J'ai aussi connecte dans la fenêtre toutes les fonctionnalité que mes camarades on pu implémenter et réaliser.

De plus, a l'aide du logiciel Gource, j'ai pu créer une courte vidéo montrant l'évolution de notre répertoire Git au cours de ces cinq dernier mois, nous l'avons mis a disposition sur notre site internet.

Mais je n'ai pas été seul, j'ai pu avoir de l'aide de mes camarades lorsque j'ai pu rencontrer des problèmes d'implantation et j'ai notamment pu les solliciter lors de choix cruciaux.

6.1.2 Christophe NGUYEN

Bien que les filtres réalisés n'aient pour le moment rien de concret lors de la première soutenance car le travail était principalement consacré au dessin, lors de cette deuxième soutenance, nous voyons mieux comment les filtres sont une partie à part entière de notre application car nous commençons à manipuler les images dans notre application.

Pour la deuxième soutenance, je me suis concentré sur la création des outils dans notre application. Le principal outil a été le seau permettant de remplir une surface donnée au clic. Il a fallu faire un travail de recherche plus conséquent par rapport à celui fourni lors de la première soutenance.

La dernière soutenance étant la plus intense en terme de travail, je me suis chargé d'ajouter un maximum d'outils possible pour ajouter la diversité dans notre application. J'ai optimisé le seau pour un rendu plus rapide et précis. J'ai également ajouté un outil pipette ainsi qu'un autre stylo : le surligneur. Ayant déjà utilisé un grand nombre d'applications de dessins, j'ai remarqué que très peu possédaient plusieurs formats pour la page, j'ai donc voulu ajouter ma touche personnelle qui était de mettre des pages différentes comme les quadrillages ou les partitions de musique.

En conclusion, ce projet a été une expérience très enrichissante car il m'a permis d'élargir mes connaissances et j'ai pu découvrir une utilisation différente de la librairie GTK ce qui augmentera mon champ de réflexion pour les prochains travaux à effectuer. Je pense encore avoir beaucoup à apprendre sur cette librairie. Ainsi, je ressors de ce travail avec la fierté d'avoir complété ce que nous avons décidé de réaliser en début de semestre.

6.1.3 Phu Hung DANG

Tout en faisant des recherches sur les codes qui pourraient être intéressants pour notre projet, j'ai réalisé lors de la première soutenance la fonction de sauvegarde de notre projet et aidé mon groupe sur l'interface ainsi que les outils de base comme le crayon.

Lors de la deuxième soutenance j'ai effectué un travail de recherche intensif afin de trouver des fonctionnalités manquantes au projet. Je me suis donc occupé du retour en arrière et du retour en avant, avec l'implémentations d'une Stack et en convertissant notre drawing-area en pixbuf, une idée qui s'avèrera extrêmement utile durant tout le reste du projet. C'est d'ailleurs la base de nos fonctions de filtre, qui ne fonctionnaient qu'avec une SDL_Surface, problème résolu en convertissant le pixbuf en surface SDL.

Enfin, pour la dernière soutenance, j'ai réalisé deux de nos formes, l'étoile avec 5 côtés et le polygone. Les coordonnées de ces deux formes étaient la partie complexe du code, mais une fois qu'elles sont acquises, il suffisait simplement de les relier. J'ai ensuite assisté les autres membres lorsqu'ils rencontraient des difficultés sur le code et en effectué des recherches.

En conclusion, ce projet m'a permis de m'améliorer sur l'utilisation de GTK et SDL, mais également sur mes compétences à travailler en groupe et à respecter les contraintes de temps. Avec des connaissances approfondies et une meilleure compréhension du C, je suis fier de pouvoir présenter ce projet dans son entièreté, avec ses lacunes certes, mais qui sont moindres face à ses qualités et à son potentiel.

6.1.4 Angélo EAP

Lors de ce projet, je me suis occupé principalement du site Internet de notre projet et des outils présents dans notre application.

Avant la première soutenance, je me suis occupé du site Internet de notre projet afin de pouvoir mettre en avant notre application. De plus, lorsque l'architecture du site a été terminée, j'ai pu aider les autres membres du groupe qui avaient du mal sur la détection des cliques de souris et de traçage de ligne.

Mon rôle dans ce projet a été principalement de m'occuper du site Internet. En effet, étant le seul à avoir déjà touché à du HTML je me suis donc proposé de travailler sur le Site Internet.

De plus, pour la seconde soutenance, je me suis concentré sur l'implémentation de l'écriture sur une image chargée ainsi que la sauvegarde de la zone de dessins. Pour cela, j'ai donc ajouté de nouvelles variables de surface et d'un tableau de pixels afin de récupérer la structure de notre zone de dessin ou image que l'on veut modifier.

Pour la dernière soutenance, ayant déjà terminé l'architecture du site, il m'a suffi d'actualiser les différents boutons ou images au cours de l'avancement du projet. Je me suis donc concentré sur le chargement d'une image dans la zone de dessins qui n'était pas très fonctionnelle lors de la seconde soutenance. J'ai de plus, aidé les autres sur les différents outils comme par exemple la rotation de l'image, la pipette et les formes.

Ce projet m'a permis de beaucoup m'améliorer sur l'utilisation de GTK et SDL, mais aussi sur mes compétences à travailler en groupe et à respecter les contraintes de temps. Je suis très fier du rendu final de notre projet.

6.2 Tableau de répartition des tâches

	Christophe	Léo	Phu Hung	Angélo
SDL				
Gestion des images			X	
Algorithme d'édition d'images		X		X
Algorithme d'application de filtre	X			
GTK				
Implémentation des boutons		X	X	
Implémentation des différents outils	X			X
Chargement et sauvegarde	X		X	
Graphisme				
Design de la fenêtre		X	X	
Ergonomie de l'application	X		X	
Communication				
Site de présentation				X
Rendus Overleaf	X	X	X	X
Divers				
Gestion du répertoire Github		X	X	
Compilation des fichiers		X	X	

Tâche \ Soutenance	Soutenance 1	Soutenance 2	Soutenance 3
SDL	30	60	100
GTK	30	60	100
Graphisme	50	80	100
Communication	40	70	100

TABLE 1 – Tableau des soutenances (en %)

6.3 Visualisation de l'historique de développement

Nous avons pu réaliser une vidéo montrant l'évolution du répertoire Git au cours de ces cinq derniers mois à l'aide du logiciel gratuit "Gource".

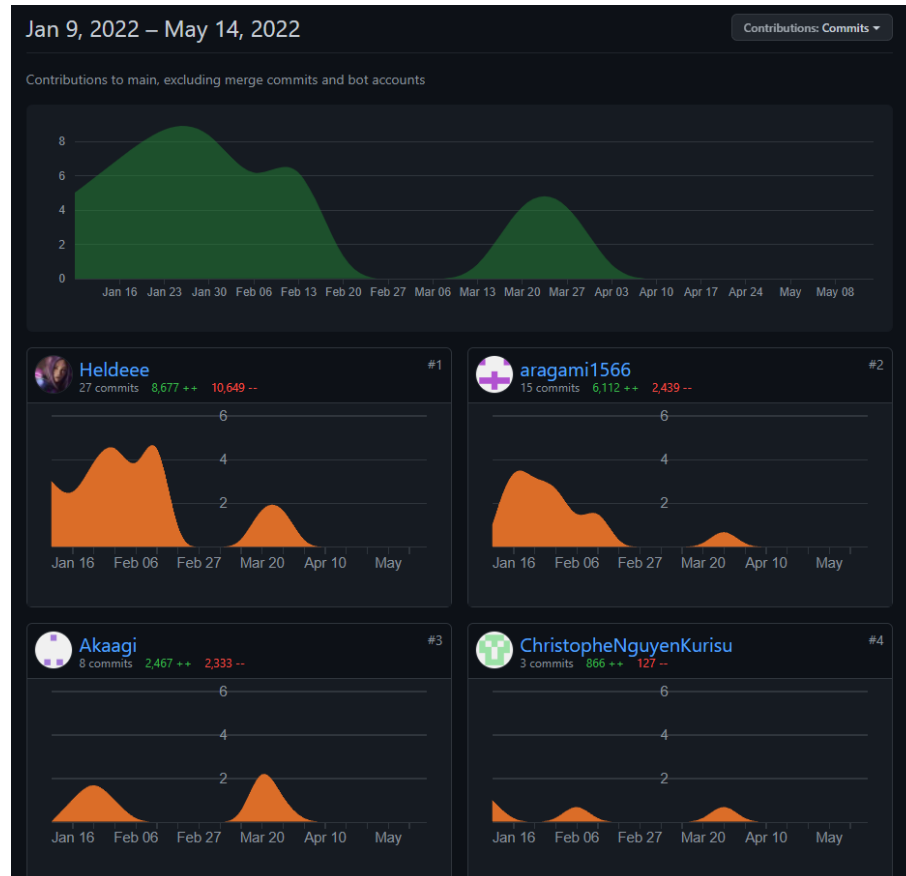


FIGURE 42 – Statistique GitHub

De plus, nous avons beaucoup travailler sur l'ordinateur du chef de projet afin de debugger et rajouter des fonctionnalités durant le projet. C'est pour cela que les chiffres peuvent être erronés.

7 Problème rencontrés et solution trouvées

Lorsque nous avons commencé le projet, nous avons voulu utiliser SDL pour la création de notre fenêtre afin de permettre l'utilisateur de dessiner. Cependant lors de nos recherches et de nos tests, SDL nous a posé beaucoup de problèmes. Nous avons donc fait d'autres recherches afin de créer cette fenêtre. C'est alors que nous avons découvert la fonction "DrawingArea" de GTK. Cependant, par la suite SDL nous sera utile pour la modification.

La détection des différents clique de souris a été un problème car nous n'arrivions pas à détecter correctement et actualiser notre fenêtre de dessin en temps réel.

L'un des problème majeur a été le fonctionnement de notre crayon. En effet, nous étions bloqué sur un fonctionnement ou le crayon n'affichait que des pixels un par un. Puis nous avons trouve une solution a notre problème grâce a une toute autre méthode qui consistait a tracer des lignes entre chaque point que le crayon détectait. Il a donc fallu mémoriser l'emplacement de la souris pour après relier les points.

La disponibilité des différents membres du groupe a aussi été compliqué car nous n'avions pas les même horaires. Nous nous sommes donc repartie des taches à faire et travailler en groupe lorsque l'on pouvait.

La fonction d'actualisation à été créer pour régler un problème que nous avons rencontrer lors de la modification de l'image. En effet, lors des différentes modification de la surface, la nouvelle surface créer n'était pas afficher sur la zone de dessin pour palier ce problème nous avons donc ajouter cette fonction.

8 Avance et Retard

La majorité des fonctionnalités que l'on a voulu réaliser sont fonctionnelles. Il nous manque néanmoins certains outils comme la loupe et le rogner, ce qui est assez embêtant mais ne nous empêche pas de présenter notre projet.

9 Avis et commentaires du groupe

Il n'y a pas eu de problèmes au sein du groupe. Le groupe est bien soudé, nous essayons le plus possibles de travailler en groupe cependant un membre du groupe n'étant pas dans la même classe que les autres cela rend un peu difficile de travailler ensemble car nos horaires diffèrent. Nous essayons néanmoins d'être présent sur un vocal Discord afin de communiquer le plus possible sur les différents avancements et problèmes.

10 Anecdotes

Lors de l'approche de la soutenance finale, trois de nos membres ont décidé de se réunir chez Phu Hung Dang afin de terminer le projet ensemble. Il s'agissait de Leo Devin et Angelo Eap, Christophe Nguyen était occupé à ce moment de l'aventure. Notre projet a beaucoup évolué durant ces trois jours. Cette productivité surprenante a permis au projet d'avancer de manière significative, et nous a permis de résoudre un grand nombre de problèmes et d'ajouter de nouvelles fonctionnalités comme les formes et la complétion du seau.

11 Conclusion

En conclusion, nous avons finalement une application qui correspond à nos attentes, avec les outils que l'on avait prévu de faire et une interface qui nous correspond. De plus, cet ambitieux projet va nous permettre de découvrir les coulisses de la gestion et de la création d'un outil d'édition d'image tout en aiguisant nos compétences informatiques et artistiques. Les contraintes liées au respect des deadlines et du travail en groupe vont grandement développer notre rigueur. Cela nous offre une expérience de travail en groupe qui nous servira dans nos futurs projets.