

Rapport de soutenance 1 - Projet S4

PiCsel

Avril 2021



HU David
MARTINS DE CARVALHO Alexis
SÉGARD Rémi
LARGENT Jean-Pierre

Table des matières

I	Introduction	3
1	Remise en contexte	3
2	Attentes collectives	3
3	Entrée en matière	4
II	Avancées	5
1	Manipulation des pixels	7
2	Interface du logiciel	7
3	Les calques et images d'animations	8
4	Importation	11
5	Exportation	11
6	Site Web	12
III	Difficultés rencontrées	14
1	Correspondance SDL-GTK	14
2	Dessin au sein du cadre	14
3	Importation d'image	15
IV	Réalisations à venir	16
1	La palette de couleurs	16
2	Les outils	16
3	Gestion de la transparence	16
4	Sélection dans l'image	16
5	Gestion des Gifs	17
6	Sauvegarde	17
V	Répartition des tâches et Planning	18
1	Répartition des tâches	18
2	Planning de l'avancement du projet	19
VI	Conclusion	20

I Introduction

1 Remise en contexte

Il y a maintenant près d'un mois et demi le groupe PiCsel s'est formé et s'est donné pour objectif réaliser un logiciel d'édition d'image (toutefois centré sur et optimisé pour la création d'images à basse définition / pixel-art). Nous avons discuté de ce à quoi devait ressembler la version finale du logiciel et quels seraient les avantages d'un tel logiciel, ainsi qu'un passage en revue de ses diverses utilisations possibles.

On a en outre fixé des objectifs en vue de la première soutenance, en plus d'avoir pensé à la façon dont nous allions amorcer la réalisation du projet.

2 Attentes collectives

Pour cette première soutenance, nous nous sentons tout d'abord enthousiastes car nous avons traversé les premières étapes de ce qui est pour nous, un tout nouveau type de projet : un logiciel, qui est plus est, d'édition d'image. Cela se démarque de précédentes réalisations pour lesquelles l'aspect logiciel était secondaire.

Désormais, cet élément occupe une place centrale dans le cadre de ce projet, ce qui implique pour nous de nouvelles approches/méthodes de travail. Par conséquent, nous sommes satisfaits de pouvoir acquérir de nouvelles connaissances en programmation .

Ensuite, outre le côté apprentissage, accompagné de nouveaux concepts à explorer comme mentionné plus haut, il faut

ajouter le fait que développer un logiciel pour la première fois nous donne une certaine fierté, ce qui vient, bien sûr, ajouter du positif à nos attentes.

3 Entrée en matière

Nous sommes à présent sur le point de passer en revue nos avancées au stade de cette première étape du développement de PiCsel, en mentionnant au passage les difficultés rencontrées (et aussi les moyens entrepris pour les surmonter).

Nous mentionnerons enfin nos prévisions pour la suite du projet, et ferons une courte rétrospective sur ce qui a été fait. Commençons à présent par décrire les avancements que nous avons apporté à PiCsel.

II Avancées

Commençons par les trois images ci-dessous. Elles rendent compte des diverses avancées réalisées à plusieurs stades du développement. On constate principalement que la manipulation au pixel par pixel est apparue peu à peu, n'étant pas présente dès le début.

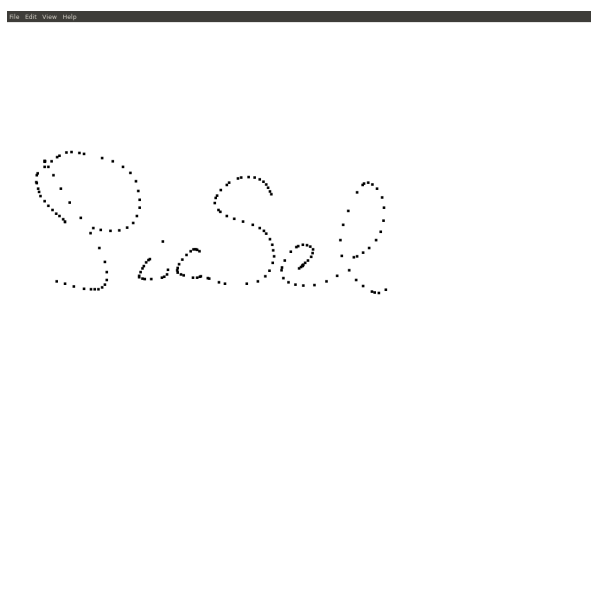


FIGURE 1 – Première version de PiCsel

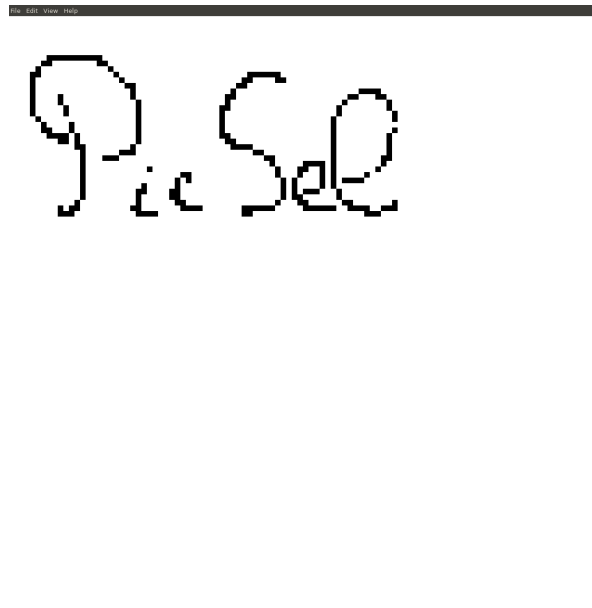


FIGURE 2 – version de PiCsel après l'implémentation de la grille

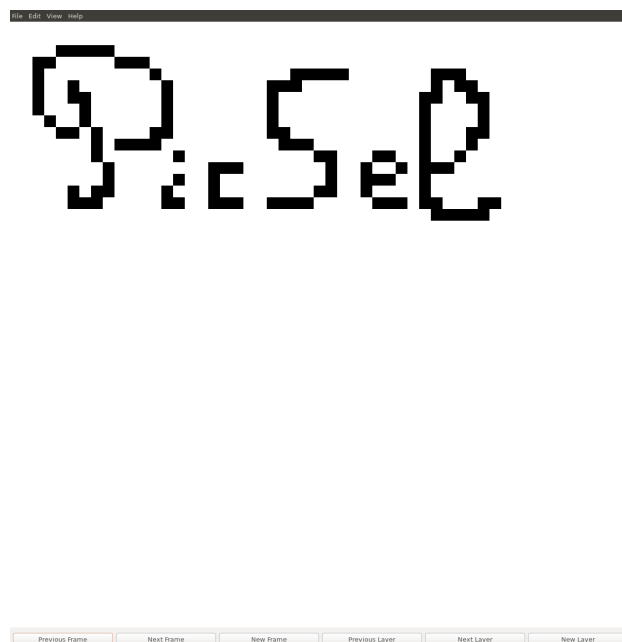


FIGURE 3 – version finale de PiCsel pour cette soutenance

1 Manipulation des pixels

Notre projet étant basé sur le dessin et l'édition d'image, il était nécessaire d'avoir la possibilité de dessiner. Nous avons donc commencé par implémenter une forme de dessin très simple. Pour la première implémentation, il suffisait de cliquer sur la surface blanche, afin de faire apparaître un carré centré sur la position du curseur.

Toutefois, cela ne correspondait pas à nos attentes, étant donné que cette fonctionnalité (Figure 1) permettait de dessiner, mais pas de faire du pixel-art à proprement dire. Nous l'avons donc améliorée en implémentant une grille non-affichée qui colore les pixels formés par la grille, en fonction de l'emplacement du curseur (Figure 2 et 3). La taille des cases de cette grille dépend de la taille de l'image ainsi que de la taille de la fenêtre.

2 Interface du logiciel

Pour l'instant, notre interface est très simple. Nous avons un grand espace blanc pour dessiner. Ce dernier correspond à la "drawing area" (espace pour dessiner). De plus, il y a un menu à onglets déroulants, mais pour le moment, la seule action réalisable depuis le menu est l'exportation d'images avec "Export".

Il y a également six boutons en bas de l'interface. Trois sont dédiés à la gestion des calques (layers). Ces trois actions sont : se déplacer au calque précédent, au suivant, et créer un nouveau calque. Les trois autres boutons performement exactement les mêmes actions, mais cette fois, ils concernent la gestion/navigation entre images de l'animation (frames).

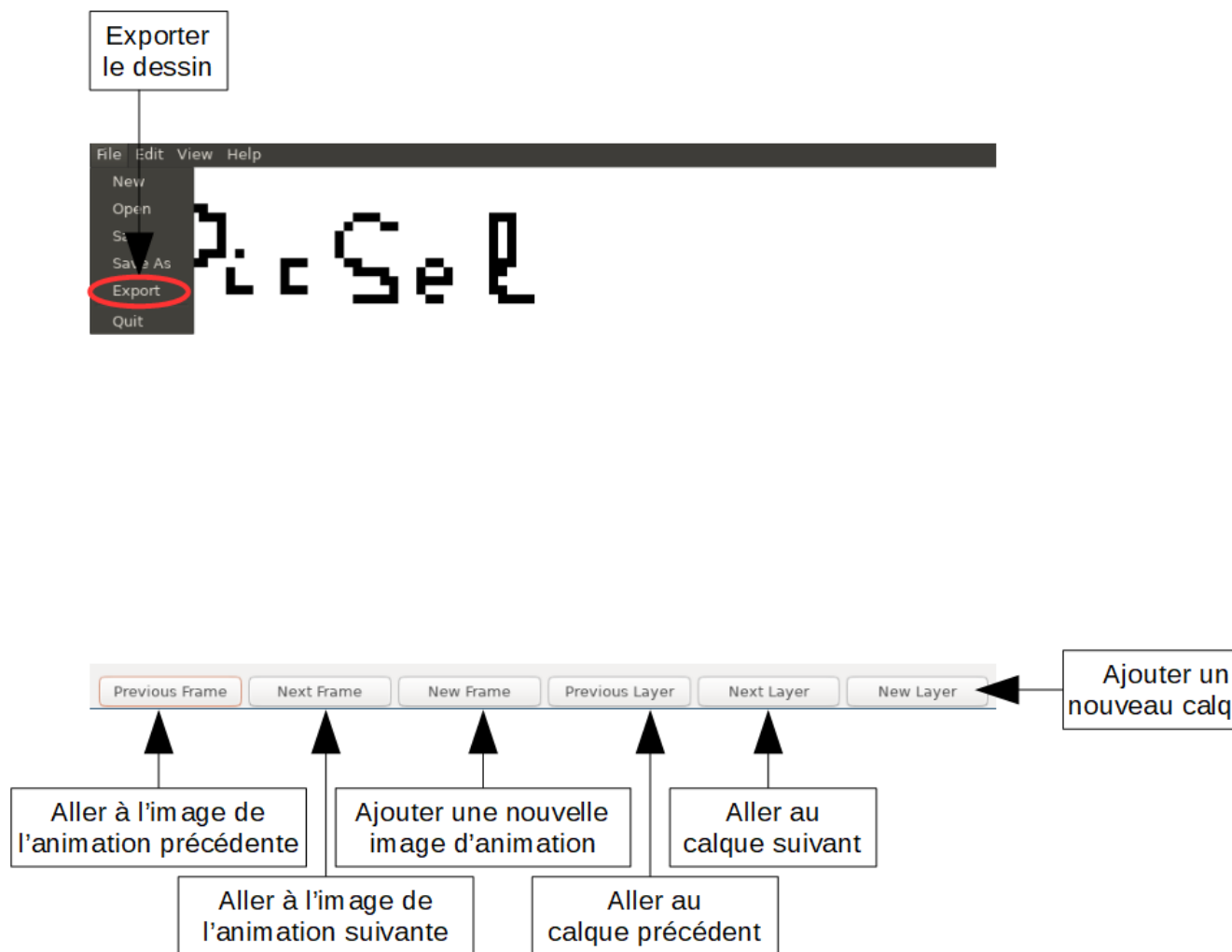


FIGURE 4 – L’interface du logiciel

3 Les calques et images d’animations

Pour implémenter les calques et les images de l’animation, nous avons créé deux structures. Les deux sont des listes chaînées avec une sentinelle. Chaque image d’animation de la liste d’images d’animation va pointer vers la prochaine, et contient elle-même sa propre liste de calques. Chaque calque de la liste contient une image qui correspond au dessin de ce calque précis.

De plus, chaque image d'animation contient une image correspondant à tous ses calques superposés. Cependant, nous n'avons pas encore géré la transparence, donc les calques n'ont pas vraiment d'utilité.

Pour la structure des images d'animation, nous avons également ajouté une variable qui va gérer le temps que va occuper chaque image au cours de l'animation. Enfin, chaque image d'animation doit avoir une liste de calques de même taille. Cependant, cette variable n'est pas très utile pour le moment, étant donné que nous n'avons pas encore d'animation.

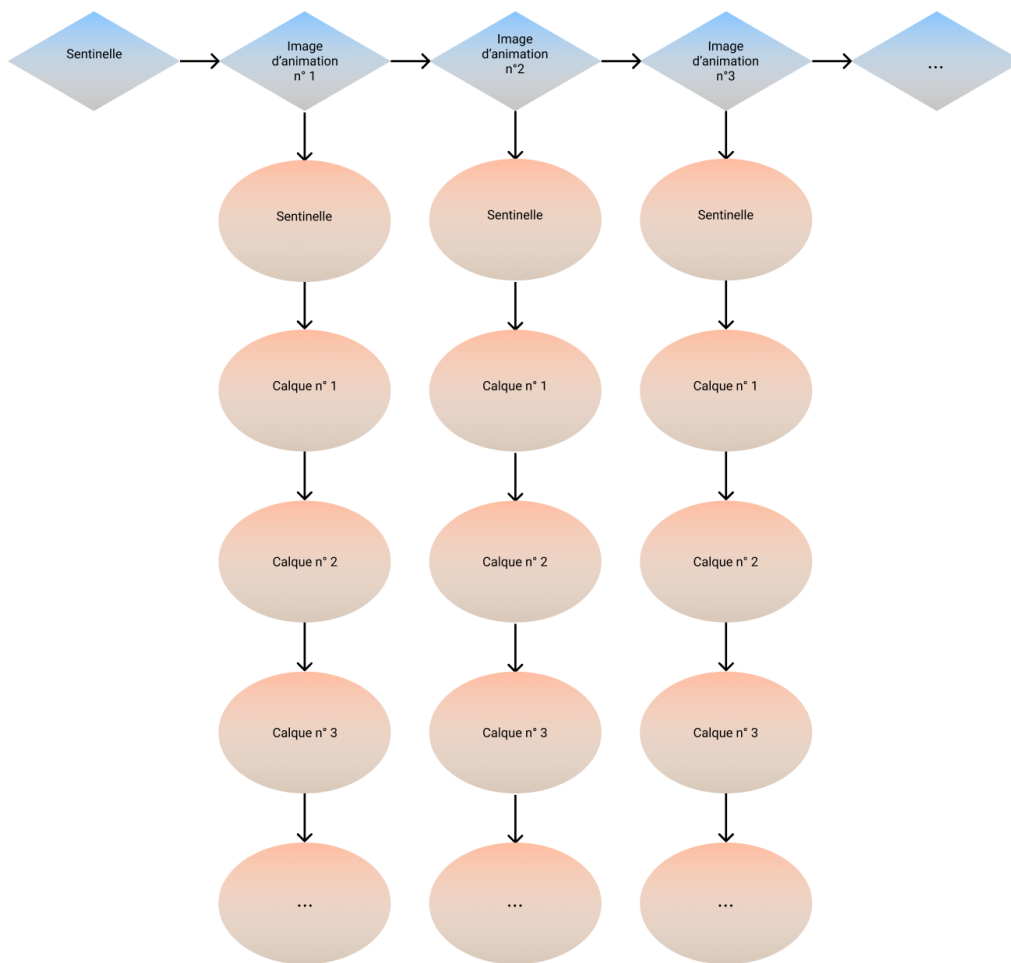


FIGURE 5 – Représentation des listes chaînées

4 Importation

Nous avons implémenté la fonctionnalité permettant l'importation des images. Pour cela, il faut directement taper dans le terminal `./main` avec le nom de l'image que l'on souhaite importer.

A terminal window with a dark background and light-colored text. The text displayed is `./main image.png`.

FIGURE 6 – Ligne de commande pour importer une l'image "image.png"

5 Exportation

Une fois qu'on a fini de dessiner, nous pouvons exporter ce qu'on a fait sur PiCsel avec le bouton "Export" dans l'onglet "File". Cela va sauvegarder l'image au format Bitmap.

6 Site Web

Nous avons crée à l'occasion de cette première étape de projet un site web, qui permet de donner un aperçu du projet PiCsel. La page d'accueil contient pour l'instant un petit message d'accroche, ainsi que le logo de PiCsel plus bas. Ensuite, si on déplace la souris vers la partie gauche, on a la possibilité de naviguer entre les pages qui composent le site. Celles-ci sont aux nombre de trois (quatre si on compte la page d'accueil).



FIGURE 7 – Page principale du site PiCsel

La page "Team" contient les présentations (en anglais), des quatre membres de l'équipe. Ensuite, la page "Timeline" est destinée à cataloguer les diverses avancées appliquées au projet. Pour le moment, on a juste affiché celles correspondant à la première étape du projet ("First Deadline" en anglais). Enfin, la page Downloads (téléchargements) contient de multiples liens. Tout d'abord, il y a des liens vers les sites officiels des bibliothèques utilisées (GTK et SDL). Ensuite, nous avons ajouté un lien vers la page GitHub de notre projet. Finalement, le présent rapport y est disponible en téléchargement .

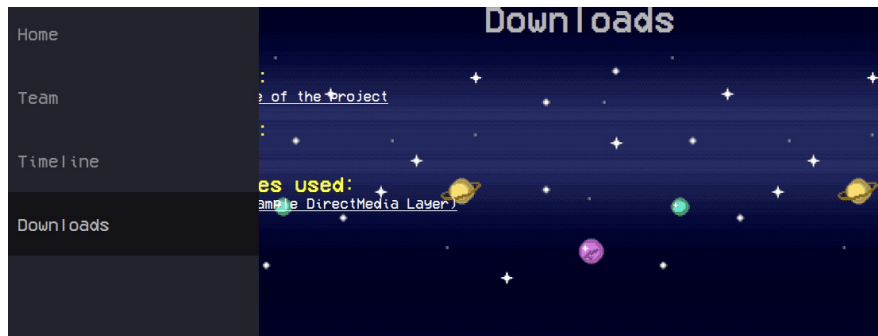


FIGURE 8 – Menu de navigation du site

III Difficultés rencontrées

1 Correspondance SDL-GTK

Pour pouvoir stocker les dessins, nous avons décidé d'utiliser SDL (Simple DirectMedia Layer). Pour chaque pixel dessiné sur GTK, un pixel sera dessiné sur l'image correspondante. Cette image pourra ensuite être traitée, notamment pour l'importation et l'exportation du dessin.

Le problème rencontré est le suivant : la taille des pixels affichés dans GTK est différente de celle affichée dans SDL. Le pixel sur GTK sera affiché en plus grand, alors que sur SDL, le pixel sera de taille identique à celle d'un pixel de l'image.

Pour régler cela, nous avons fait un calcul simple, avec un modulo, pour que GTK affiche les pixels à la bonne taille. Ainsi, il dessinera sur une "grille", dont les cases correspondent aux pixels de l'image.

2 Dessin au sein du cadre

Pour que l'application se comporte comme il le faut, nous devons faire en sorte qu'un clic en dehors de la fenêtre ne soit pas traité. Ce problème est réglé en comparant la position du clic et la taille de l'image (et non pas la taille de la fenêtre). En effet, la taille de l'image éditée est souvent plus petite que l'espace de dessin de GTK, puisque les dimensions d'un pixel sont constantes (les pixels au bord de l'espace de dessin ne peuvent pas être "étirés" pour remplir l'espace inutilisé).

3 Importation d'image

Pour importer des images, nous utilisons une fonctionnalité de SDL. Malheureusement, le comportement de cette fonction n'est pas entièrement celui désiré. En effet, d'après nos essais, le format dans lequel l'image est stockée peut parfois différer de celui que nous créons pour stocker les images des calques. Pour dessiner l'image d'une animation (c'est-à-dire que l'on fusionne les calques), on recopie les calques un à un, dans une même surface. Or si le format avec lequel l'image est importée est différent, il est possible que la copie des pixels de l'une dans l'autre ne se déroule pas comme il se doit.

IV Réalisations à venir

1 La palette de couleurs

Pour l’instant, nous ne pouvons dessiner qu’avec une couleur, ce qui n’est pas vraiment intéressant. Nous allons devoir améliorer l’interface pour ajouter une palette de couleurs, afin que l’utilisateur ait plus d’options pour dessiner.

2 Les outils

Il faudra rajouter quelques outils pour faciliter certaines tâches de l’utilisateur. Nous voudrions par exemple rajouter une option pour dessiner directement certaines formes géométriques, comme un cercle ou un rectangle par exemple.

3 Gestion de la transparence

Comme mentionné précédemment, nous n’avons pas encore géré le cas avec les transparences. Cela va être très important, notamment pour pouvoir utiliser les calques, ce qui rajoute une fonctionnalité permettant d’avoir des dessins encore plus poussés et précis. De plus, il peut être plus intéressant de dessiner des personnages sur un fond transparent et non blanc, pour différencier les pixels non édités de ceux volontairement remplis avec du blanc.

4 Sélection dans l’image

Un outil qui va demander un peu plus de travail est la sélection. En effet, l’utilisateur aura la possibilité de sélectionner

une partie du dessin et la glisser là où il le souhaite.

5 Gestion des Gifs

La gestion des gifs sera gérée par des structures d'images d'animation. Une fois terminées, nous aurons la possibilité de les afficher et les modifier.

6 Sauvegarde

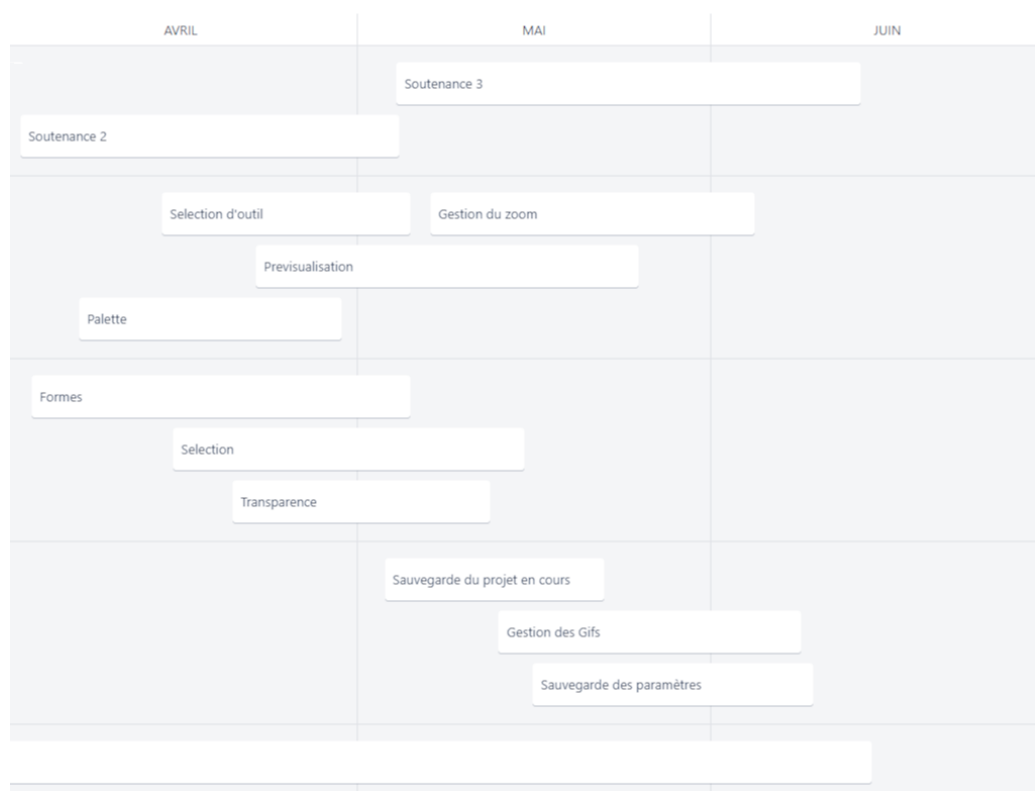
Pour améliorer la productivité de l'utilisateur, nous voudrions qu'il puisse, bien évidemment, sauvegarder ses dessins mais également sauvegarder ses paramètres favoris. Cela fera perdre moins de temps à configurer tous les paramètres, et va demander un certain effort, car le format .gif est assez unique et particulier, en comparaison avec d'autres formats "traditionnels".

V Répartition des tâches et Planning

1 Répartition des tâches

	David	Rémi	Alexis	Jean-Pierre
Interface				
Affichage de l'image	X	X		
Palette	X			X
Sélection d'outil	X			X
Sélection du calque		X	X	
Sélection de l'image dans l'animation		X	X	
Prévisualisation			X	X
Gestion du zoom		X		X
Manipulation de l'image				
Dessin	X	X		X
Sélection		X		X
Formes	X		X	
Calques		X		X
Diaporama		X	X	
Transparence			X	X
Fichiers				
Importation/Exportation	X		X	
Gestion des Gifs		X	X	
Sauvegarde du projet en cours	X		X	
Sauvegarde de paramètres	X			X
Site Web				
Site Web		X	X	X

2 Planning de l'avancement du projet



VI Conclusion

Pour cette première étape de projet, nous avons réalisé de nettes avancées. Même si toutes les fonctionnalités que nous avions prévues pour l'heure ne sont pas totalement fonctionnelles, nous sommes fiers de ce que le logiciel est déjà capable de faire.

En effet, même si les difficultés rencontrées au niveau des structures des calques nous ont ralenti sur la gestion des calques, ainsi que sur l'animation des images, nous avons réussi à implémenter le dessin, ainsi que l'importation et l'exportation des images.

Enfin, nous restons enthousiastes pour la suite du projet. Il nous tarde de réaliser et utiliser les futures fonctionnalités qui renforceront PiCsel.