Roberto NIGHTINGALE CASTILLO

Patrice GALLANT

Projet synthèse

420-C61-VM

Groupe 00001

C’est fou. Riez !

Document de conception

Travail présenté à

Monsieur Jean-Christophe Demers

Cégep du Vieux Montréal

Départements d’informatique

2 mars 2023

Table des matières

[MAQUETTES 3](#_Toc128506360)

[Fenêtre de dessin 3](#_Toc128506361)

[Fenêtre de changement de mot de passe 6](#_Toc128506362)

[Fenêtre de profil 6](#_Toc128506363)

[CAS USAGE 7](#_Toc128506364)

[Aire dessin 7](#_Toc128506365)

[Changer les paramètres et préférences du profile usager 7](#_Toc128506366)

[Fil d’actualité 8](#_Toc128506367)

[Galerie 8](#_Toc128506368)

[Inscription d’un nouveau compte 9](#_Toc128506369)

[Animation flèches Fourier 9](#_Toc128506370)

[Menu d’options 10](#_Toc128506371)

[Modification du mot de passe 10](#_Toc128506372)

[Panneau de connexion 11](#_Toc128506373)

[DIAGRAMME CLASSE 12](#_Toc128506374)

[Vue 12](#_Toc128506375)

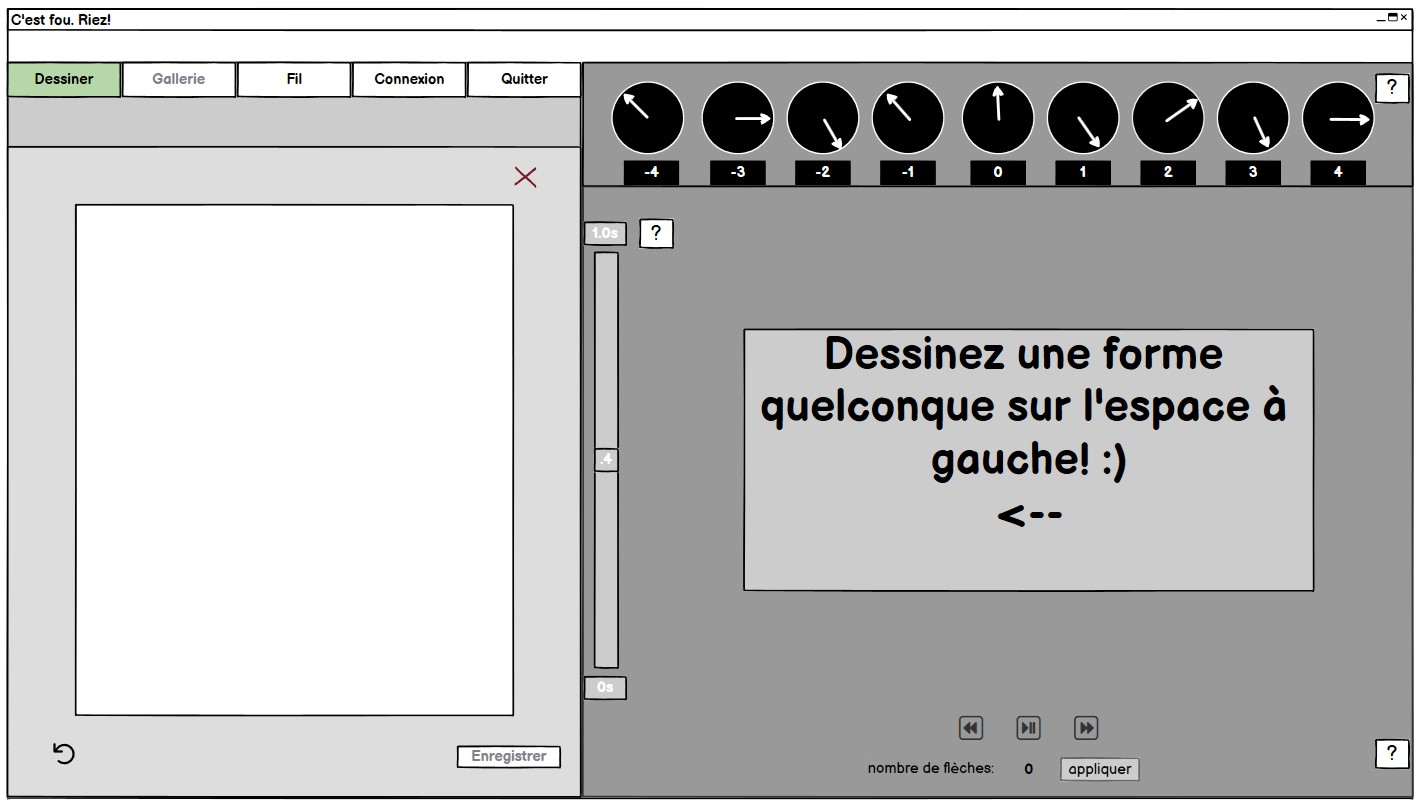
[Modèle 12](#_Toc128506376)

[Serveur/DAO 12](#_Toc128506377)

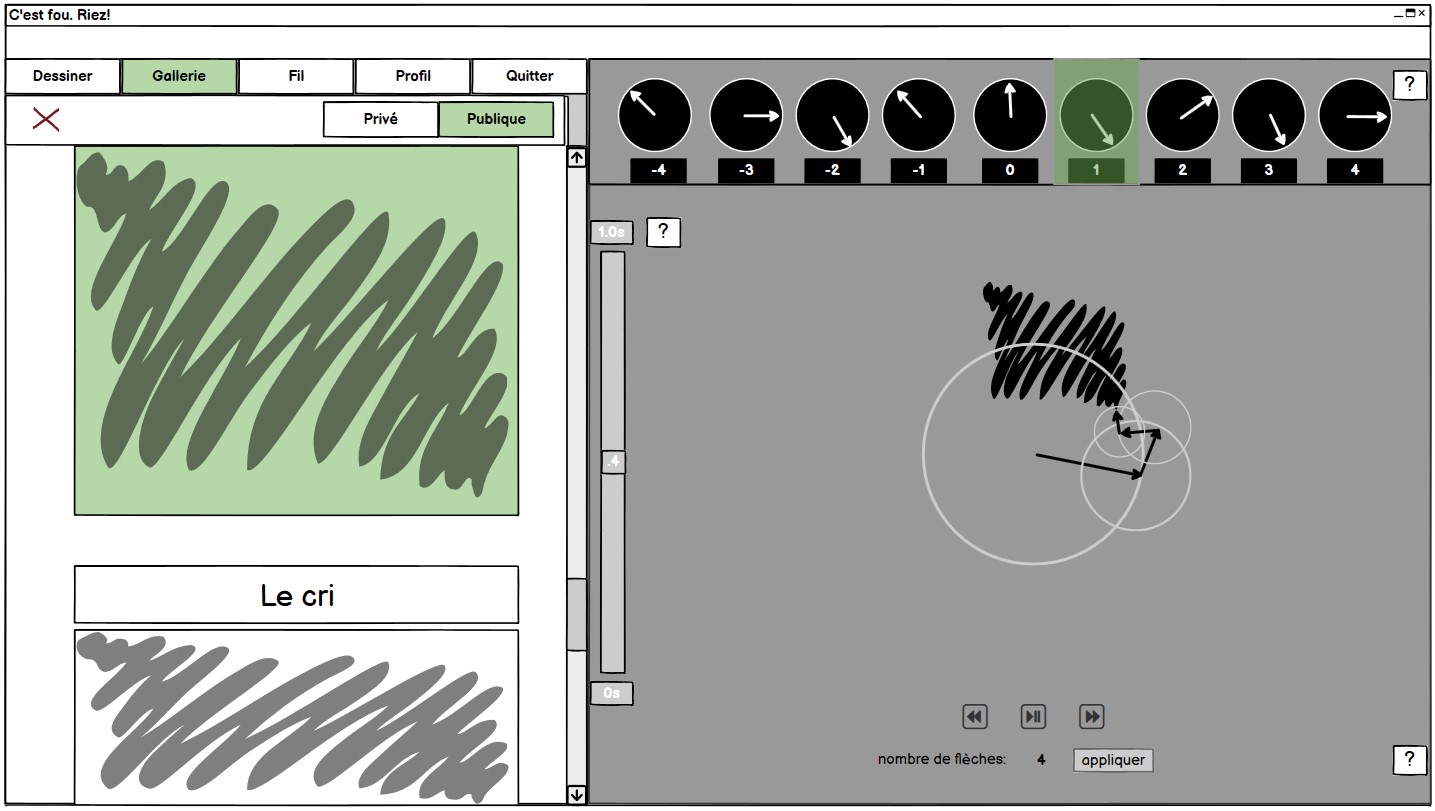
[Structure de données externe 12](#_Toc128506378)

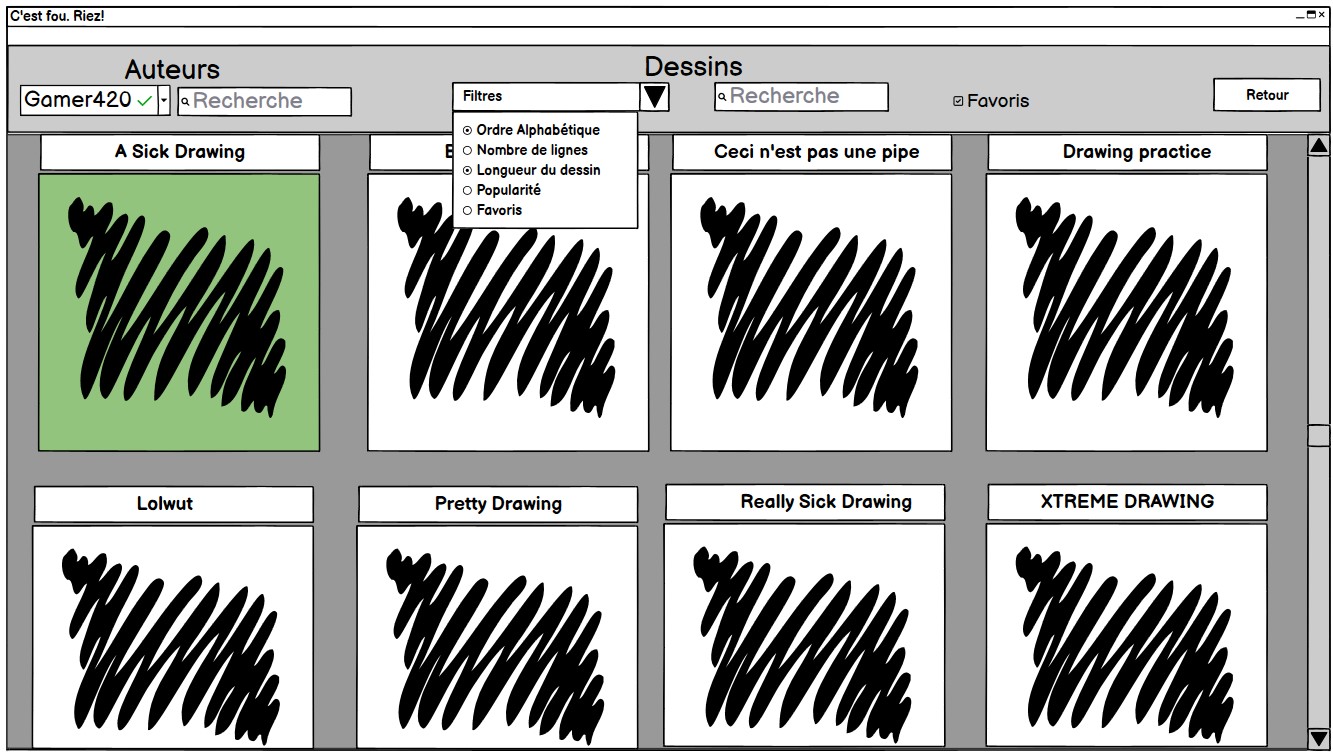
[Éléments de conception 12](#_Toc128506379)

# MAQUETTES

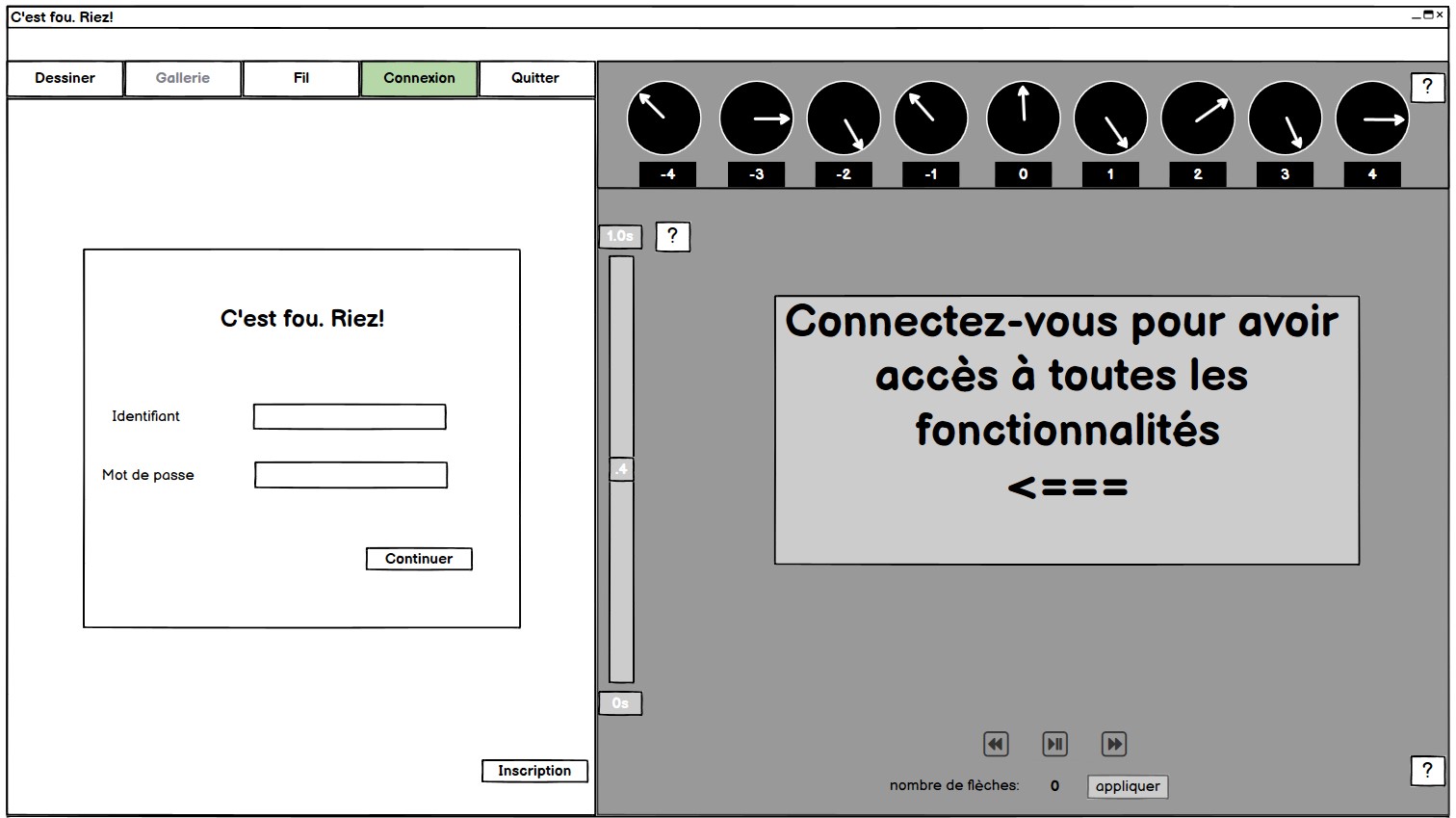
Fenêtre de dessin

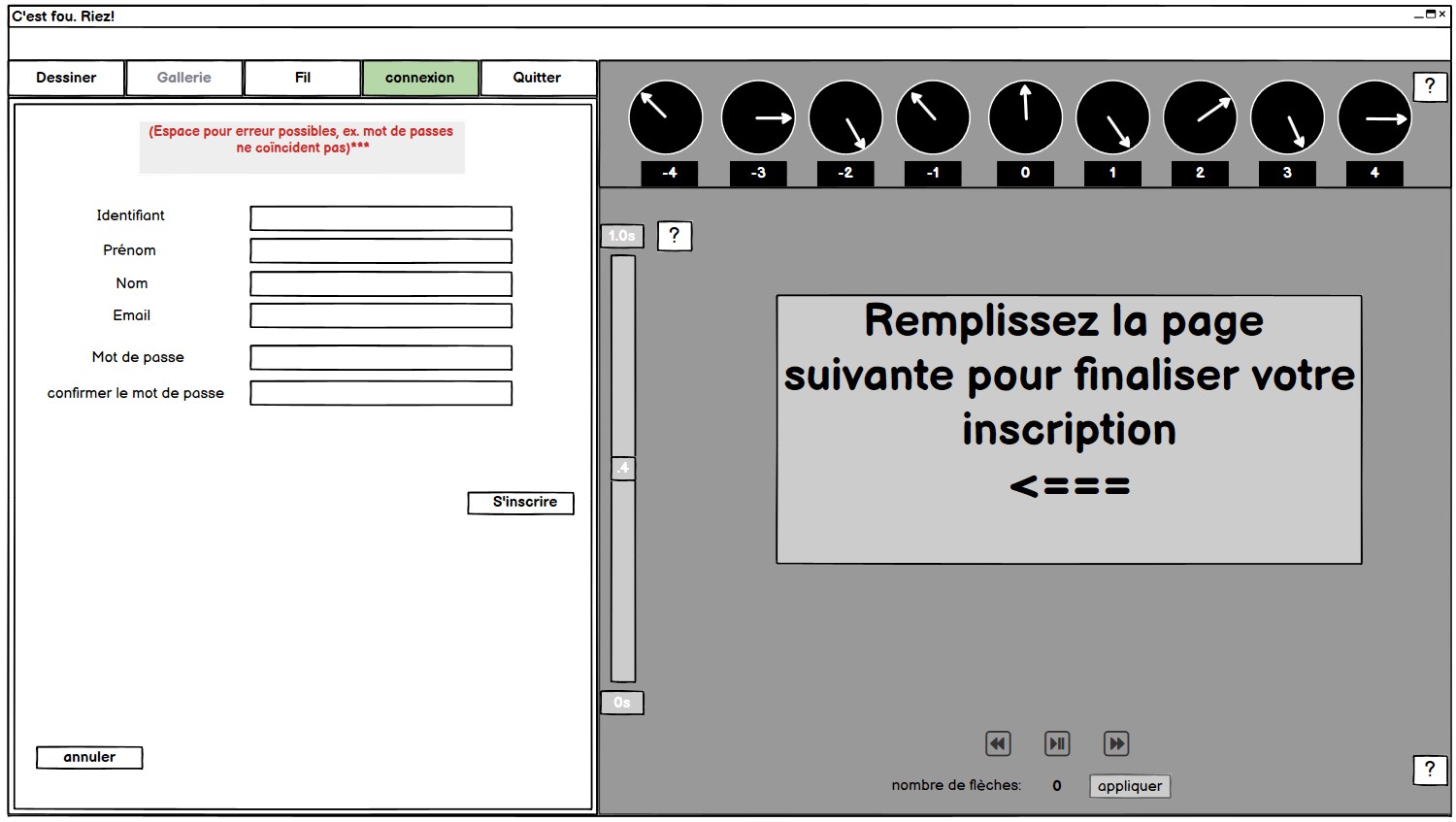
Fenêtre de galerie de dessins



Fenêtre de fil d’actualité

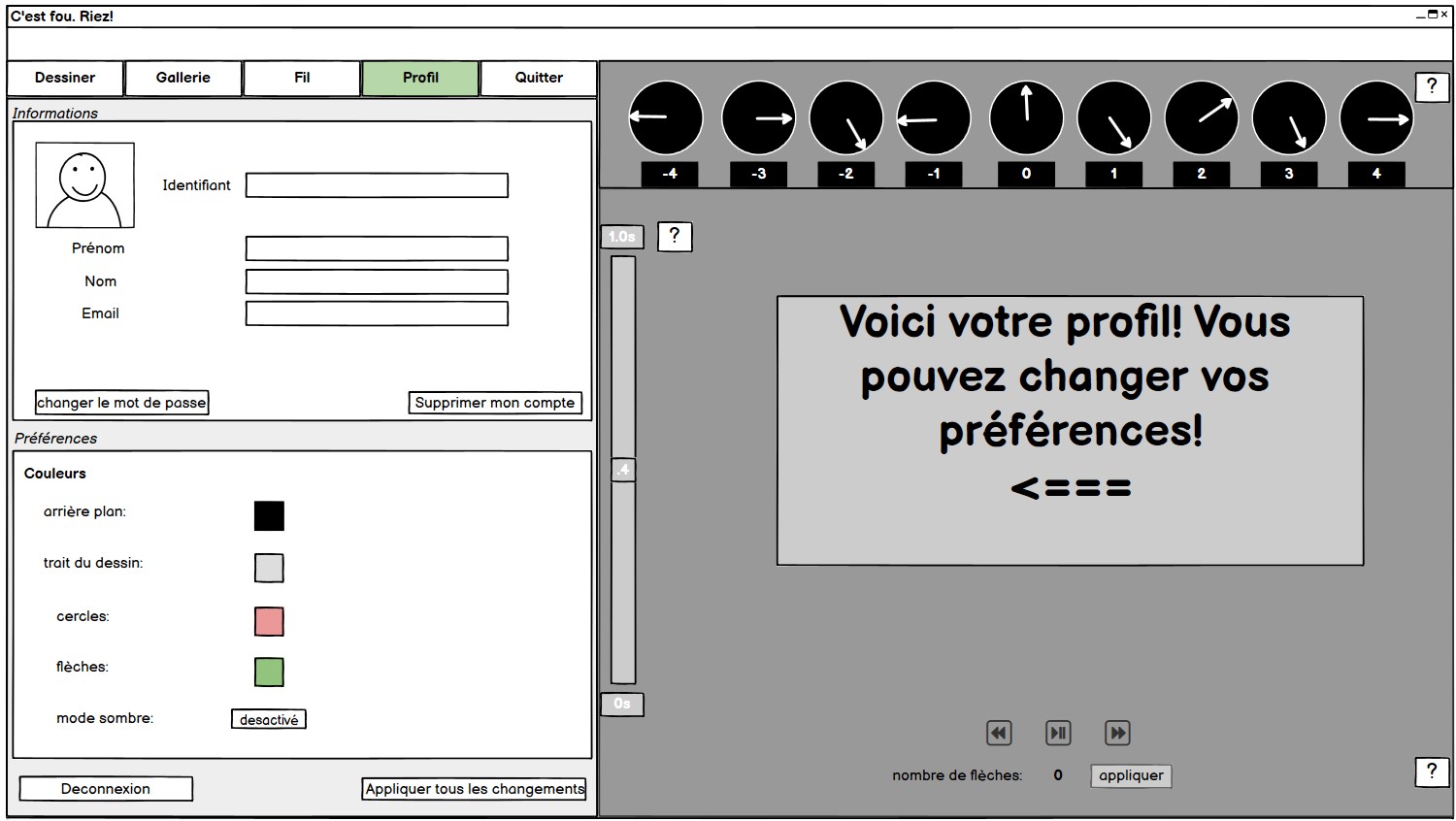
Fenêtre de connexion



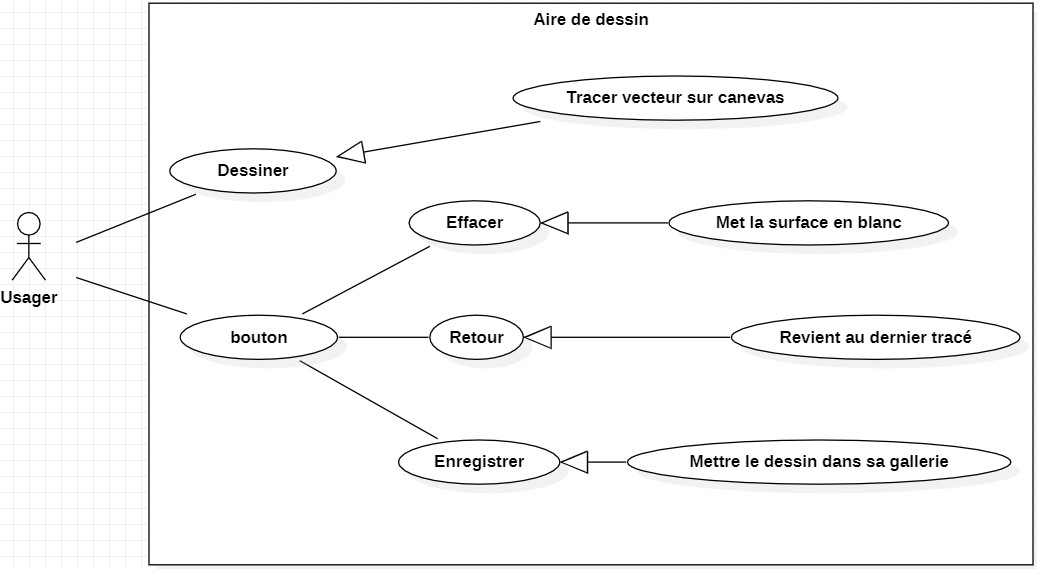
Fenêtre de création de comptes

Fenêtre de changement de mot de passe

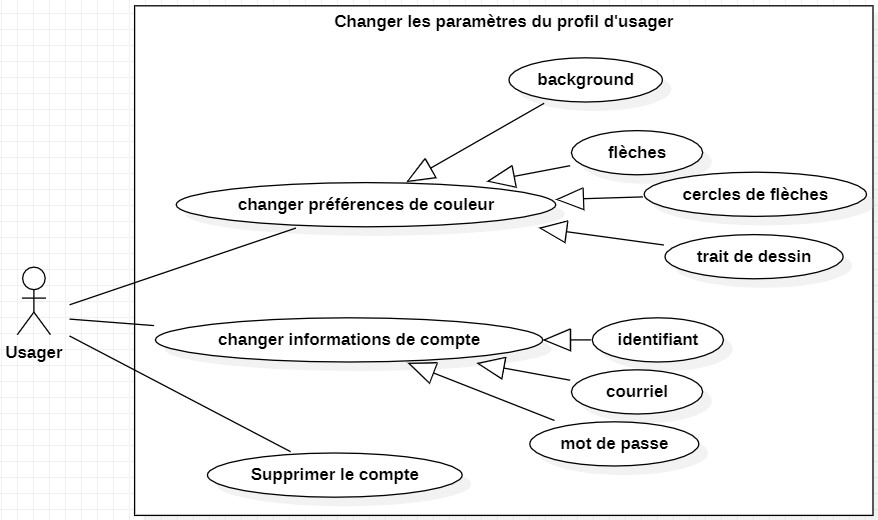
Fenêtre de profil



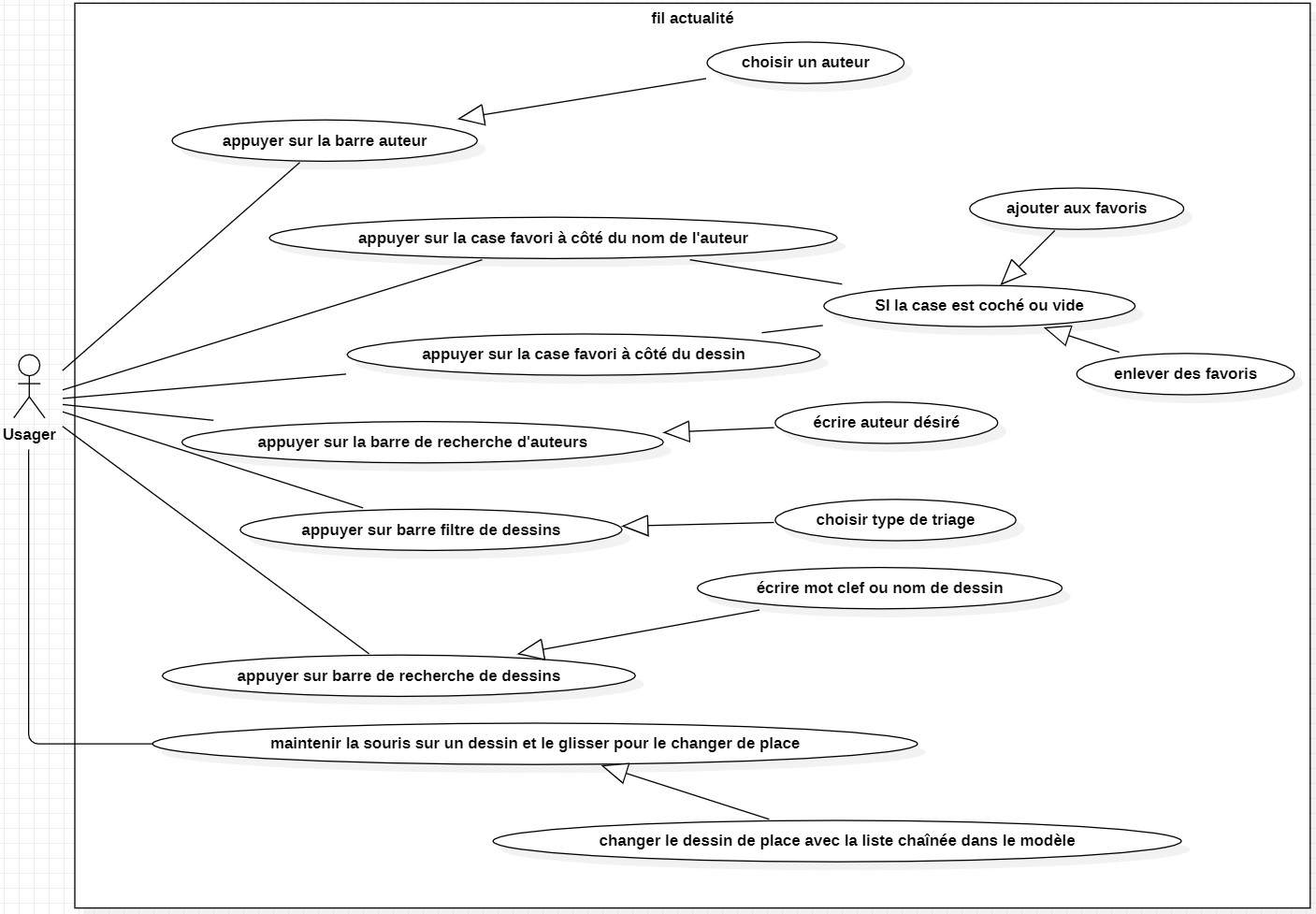
# CAS USAGE

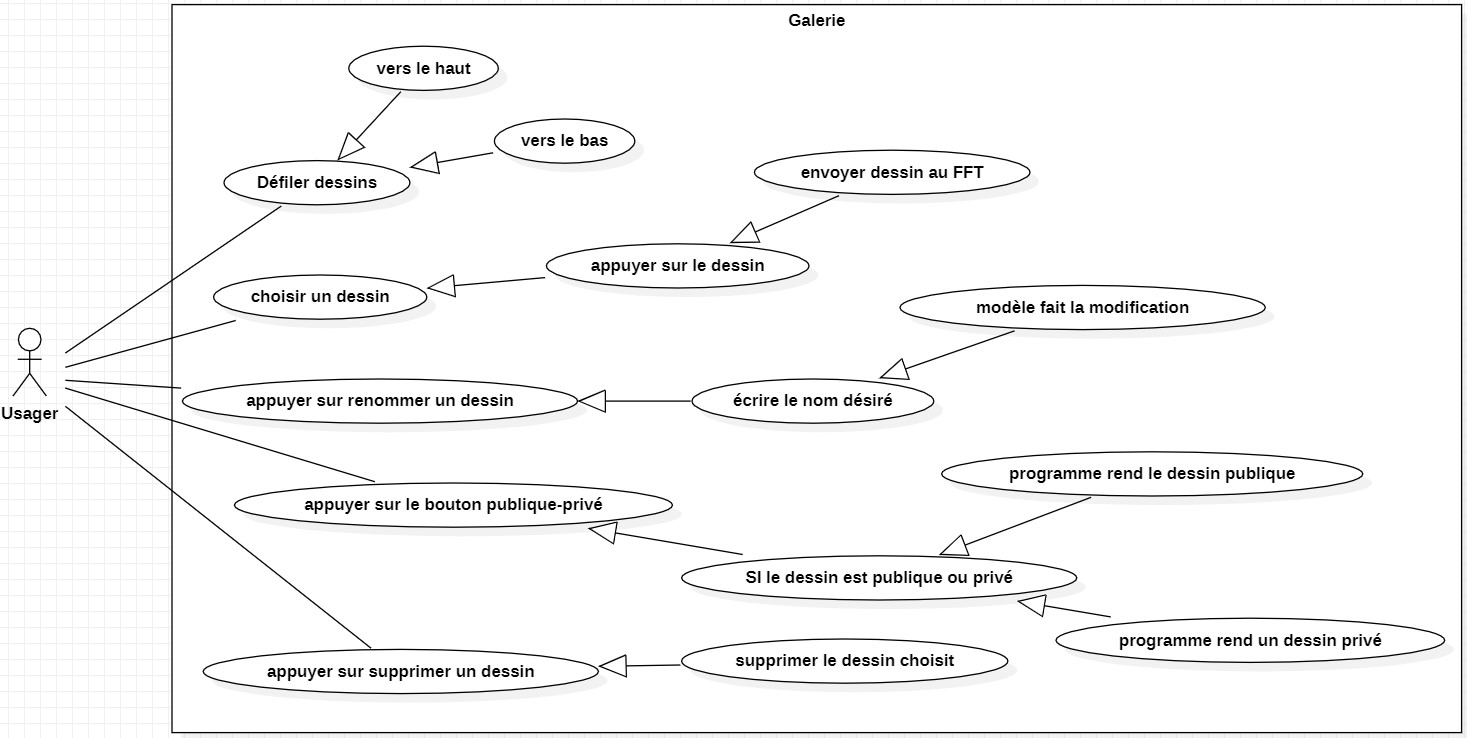
Aire dessin

Changer les paramètres et préférences du profile usager

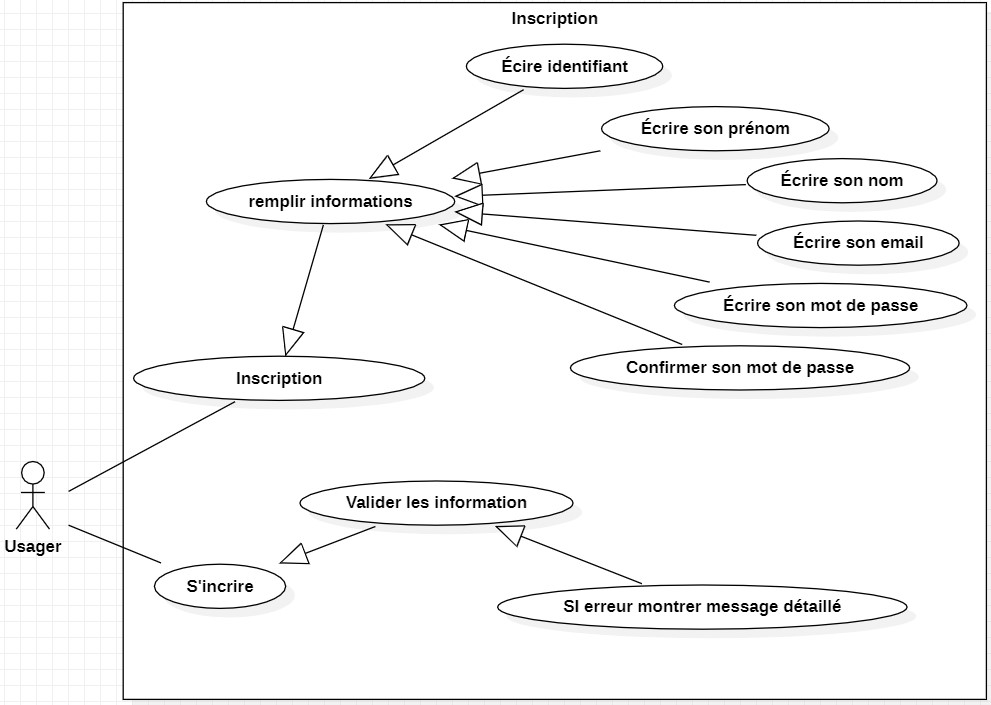


Fil d’actualité

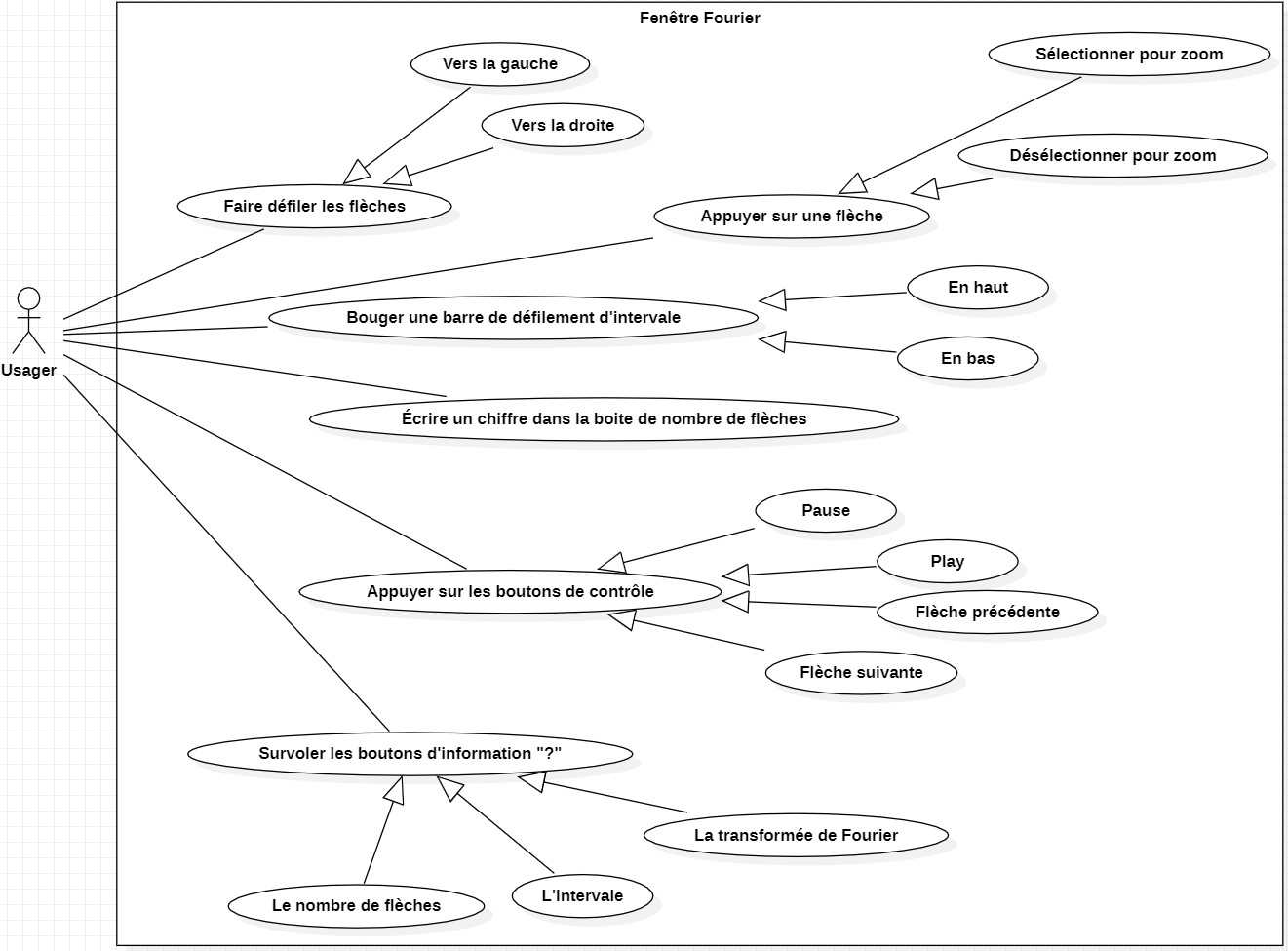


Galerie

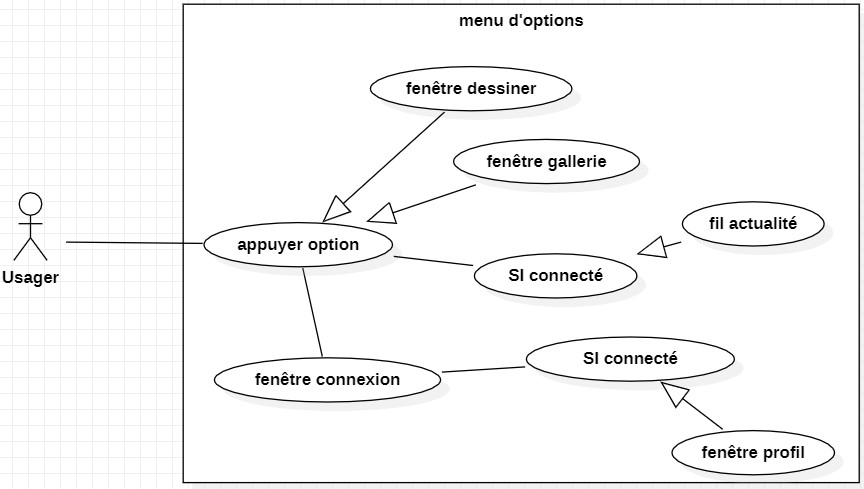
Inscription d’un nouveau compte



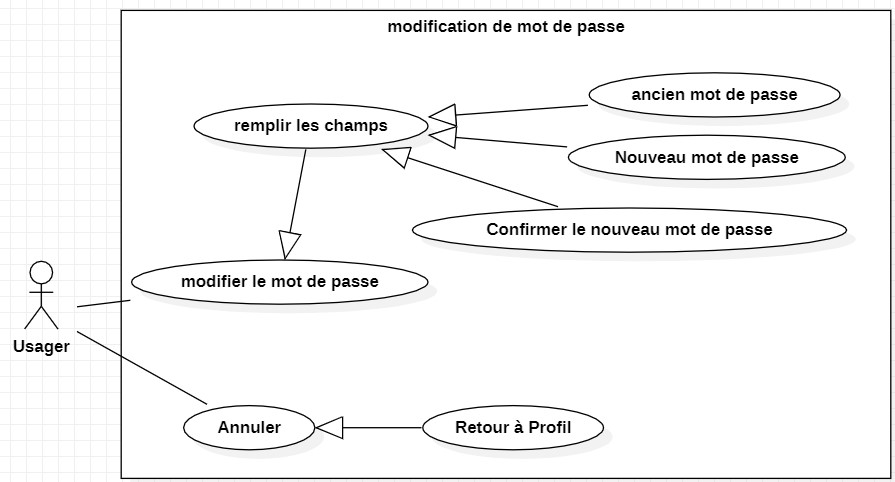
Animation flèches Fourier



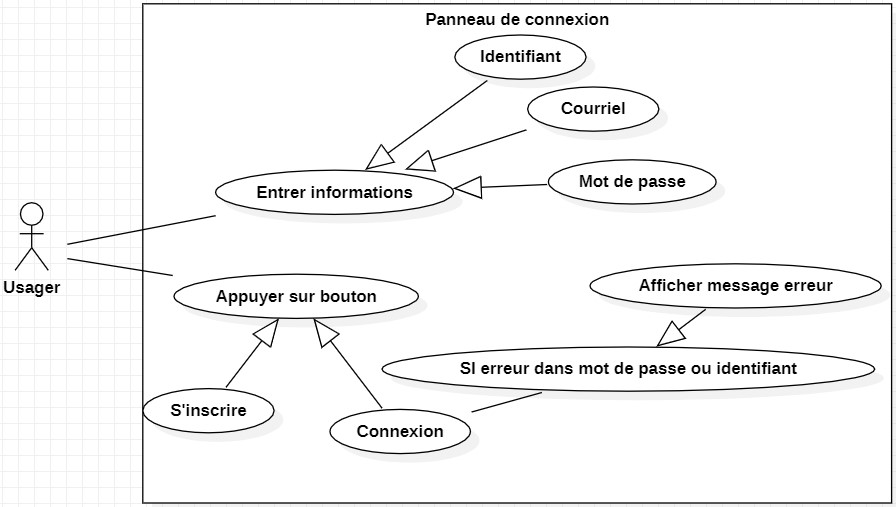
Menu d’options



Modification du mot de passe



Panneau de connexion



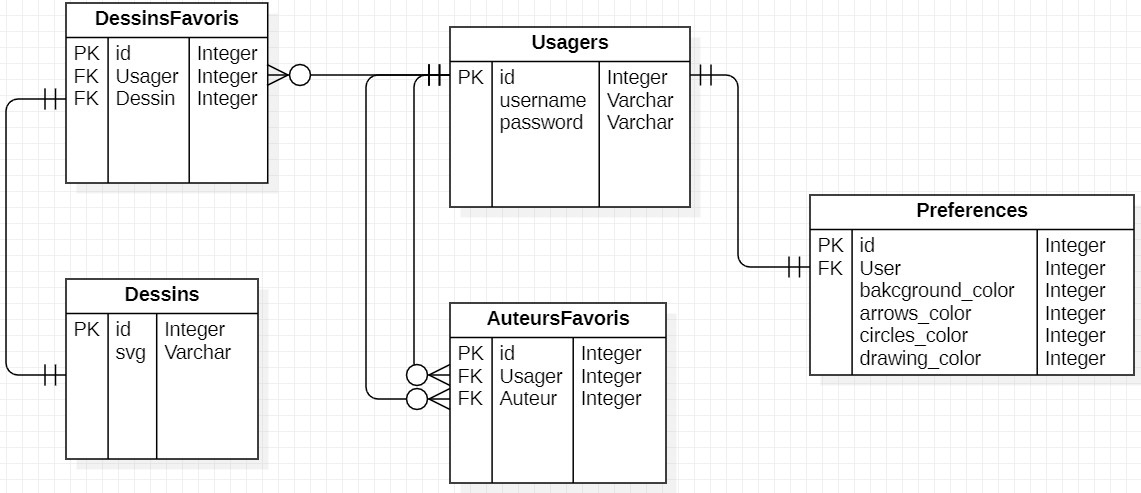
# DIAGRAMME CLASSE

Vue

Modèle

Serveur/DAO

# Structure de données externe

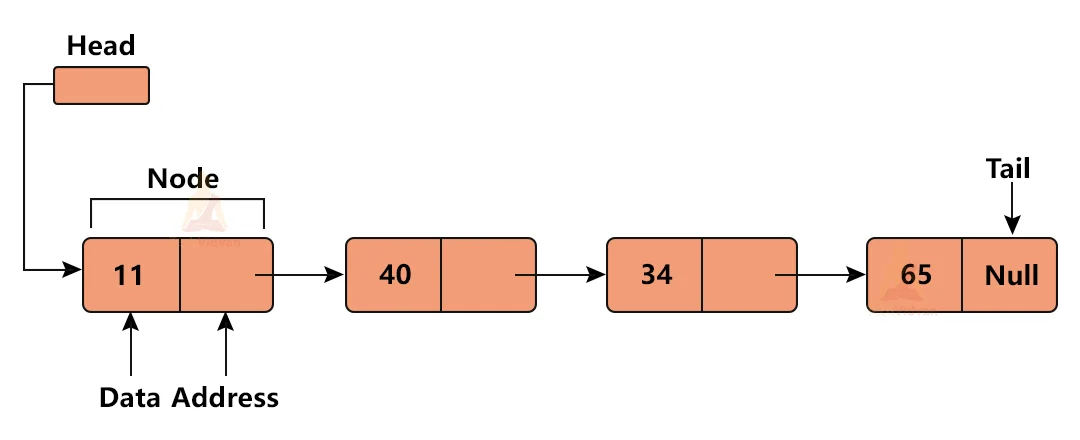


# 

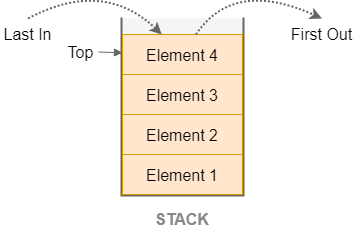
# Éléments de conception

Structures de données *(Data Structures)*

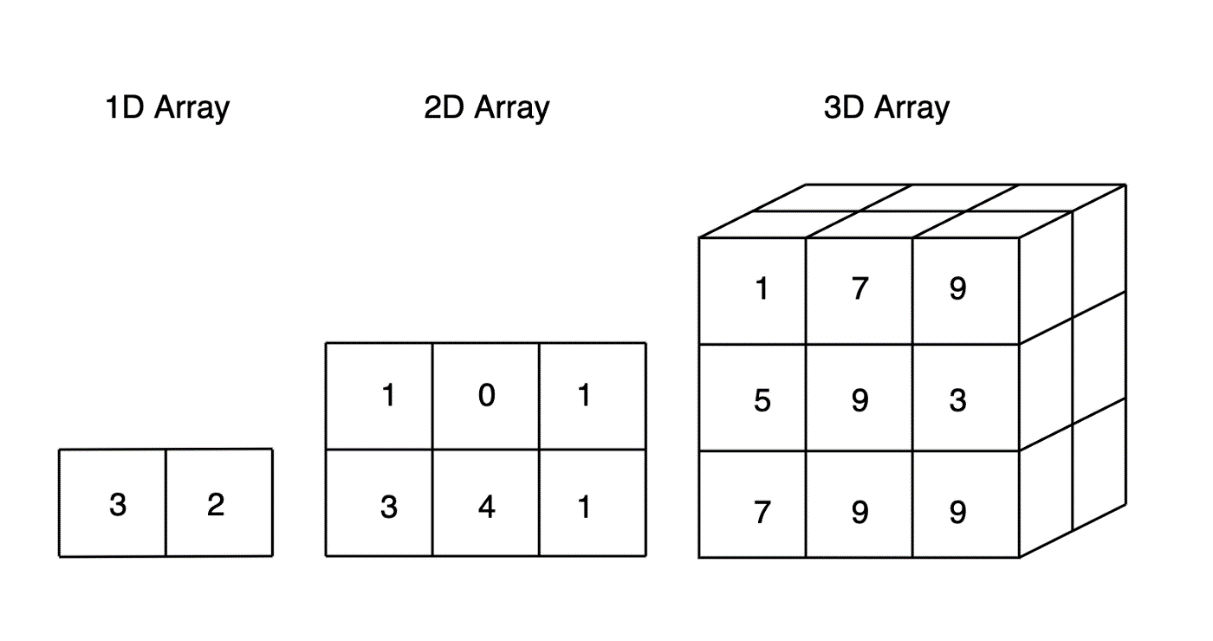
Dans le contexte de notre projet, nous allons avoir une galerie d’images exposant les dessins des différents usagers de l’application. Ces derniers pourront faire glisser des images d’un endroit à un autre en gardant la souris appuyée sur l’élément, cela afin d’organiser la galerie à leur manière. Pour accomplir cette tâche de manière optimale, nous allons faire l’usage d’une liste chainée *(Linked List).* Cette structure de donnée *(Data Structure)* est munie d’une tête *(head)* étant le premier élément de la liste et d’une queue *(tail)* étant le dernier. À l’intérieur de la structure de données, il y aura des nœuds constitués de 2 parties. La première fait référence à l’objet venant de la classe: Dessin, qui contient les informations de celui-ci et la deuxième est une référence au nœud suivant. Dans le cas de la queue, son deuxième morceau ne pointera sur rien *(None)*. Ayant maintenant ce concept en tête, il est plus simple de réordonner la liste en modifiant uniquement les pointeurs des nœuds. Cette structure de données est celle que nous comptons programmer de A à Z.



Considérant que le programme comprend une zone de dessin, nous voulons donner la possibilité à l’utilisateur de retourner à son dernier tracé au cas où il le juge nécessaire. Logiquement, il faut tenir compte de tous les traits que l’usager fait s’il veut revenir en arrière plusieurs fois. La structure de données qui nous sera utile dans ce cas est la pile *(stack).* Cette dernière suit le principe du « dernier entré, premier sorti » *(Last In First Out)*. Elle nous permet d’empiler chaque tracé par-dessus un autre et d’enlever le premier qui se trouve sur la pile, étant le dernier à avoir été empilé. Nous allons procéder de la manière suivante : lorsque l’usager va appuyer sur la souris et la glissera pour dessiner, le chemin *(path)* de son tracé sera stocké au moment où il cesse d’appuyer. Quand l’usager va appuyer sur retour, il suffira de retirer le dernier chemin dessiné et de dire à l’interface de refaire le dessin formé de ces chemins.



Au sujet de l’animation, nous voulons faire tourner les vecteurs autour de leur centre de rotation, tous en même temps. Pour que ça soit possible, les coefficients des vecteurs seront calculés à l’aide de l’algorithme de transformation de Fourier rapide. Nous allons faire usage des listes : « numpy.ndarray » de la libraire NumPy pour conserver l’angle de départ du vecteur, ainsi que sa taille. Cette structure de données est optimale pour les opérations sur des listes, qui dans ce cas, seront des matrices à deux dimensions. Dans cette situation, il faut multiplier chaque objet de la liste, étant des listes contenant les informations mentionnées précédemment, par une matrice de sinus et de cosinus. Cela nous permettra de créer un effet de rotation dans l’interface graphique.



Something something lists I guess.

Patrons de conception

Notre interface graphique sera organisée selon le patron de conception du composite. Elle sera faite avec Qt, une librairie Python faite pour la création d’interface graphique, et sera faite d’un élément principal qui sera subdivisé en plusieurs widgets, eux-mêmes subdivisés en d’autre widgets. Ces widgets seront des objets créés à partir de classes que nous aurons programmés, et donneront une structure en forme d’arbre. [Expliquer pourquoi ce patron est utile à notre travail].(behavior normal de QT + ca se struct mieux, )

Pour que nos widgets puissent envoyer des informations et des demandes d’action aux autres parties de notre programme, nous utiliseront des signaux. Ces signaux seront implémentés selon le patron de conception de chaine de responsabilité, où chaque parent de l’émetteur de signaux le recevra et décidera de faire quelque chose avec ou de le lancer plus haut dans la hiérarchie. Par exemple, un widget qui reçoit un signal de changement d’interface d’un sous-widget pourra faire l’action de changer de sous-widget, et s’il reçoit un signal de sauvegarde, il l’enverra plus haut vers le contrôleur qui l’enverra ensuite au modèle.

Nous implémenterons un patron de conception de template pour ce qui est des options de tri qui se trouvent dans la galerie d’image des autres usagers. Ces tris ont des fonctions très similaires, mais avec une variation de comment ils font le tri des images. En les traitant comme des templates d’une classe abstraite, nous avons moins de duplication inutile de code, et ce sera plus facile d’ajouter des options de tri supplémentaires si nous le désirons.

Expression régulière

La galerie de dessins de tous les usagers de notre projet va avoir deux barres de recherche, une pour les auteurs et une pour les dessins. Ces barres de recherche vont transformer ce qui sera écrit par l’usager en expression régulière et l’appliquera soit aux identifiants d’usager ou aux noms de dessin, afin de n’afficher que ceux qui contiennent la séquence entrée.

Algorithme et Mathématique

Le concept au cœur de notre projet, les séries complexes de Fourier, va être principalement exécuté par l’algorithme de Fast Fourier Transform (FFT) que nous allons implémenter. L’algorithme va prendre une série de coordonnées dérivée d’un fichier d’image SVG et va traiter ces séries de points comme des ondes. Essentiellement, c’est comme si des ondes avaient été enroulées autour d’un point sur le chemin d’un cercle. Il traitera ensuite ces ondes pour obtenir une série de nombres complexes que nous pourrons utiliser pour déterminer le rayon et l’angle de départ des vecteurs que nous aurons besoin pour recréer le dessin. Pour bien implémenter le FFT, il nous faudra un niveau raisonnable de compréhension des intervalles, des nombres complexes, de l’utilisation des sinus et cosinus dans des équations, dont la rotation de vecteurs, ainsi que de concepts plus simple comme les exposants.

Veille technologique

Le cours de veille technologique nous permet d’approfondir nos connaissances en mathématiques pour aller de l’avant dans le projet. Étant donné que Joseph Fourier était un grand mathématicien, les notions qu’il utilisait sont encore aujourd’hui considérées comme des mathématiques d’études supérieurs. N’ayant pas eu la chance d’explorer les aspects du calcul différentiel et calcul intégral en informatique, le cours de veille technologique est pour nous du temps précieux pour aller effectuer des recherches à ces sujets.