

Rapport Projet Programmation Objet Orientée 1

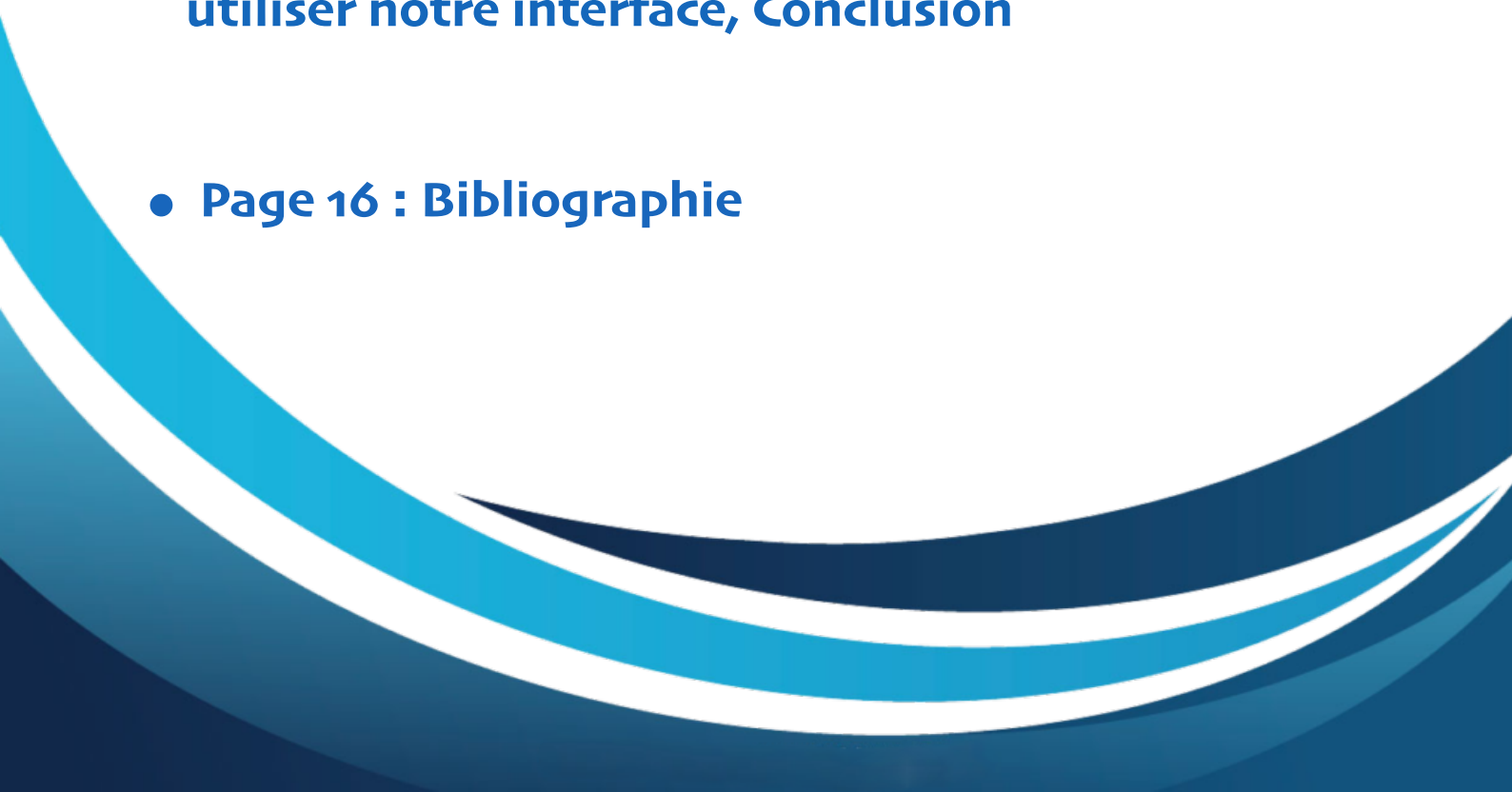
Années 2021/2022 - Licence 2

**Venditti Chloé
Achir Thinhinene**

- **Page 2 - Sommaire**




Sommaire :

- **Page 3 : But du projet**
 - **Page 4 - 9 : Étapes de la réalisation du projet**
 - **Page 10 - 13 : Problèmes rencontrés lors de la réalisation**
 - **Page 14 - 15: Résultat final du projet et comment utiliser notre interface, Conclusion**
 - **Page 16 : Bibliographie**
- 

But du projet :

Lors de notre semestre 4 nous avons du réaliser un projet pour l'UE Programmation Objet Orientée 1. Le but du projet était de créer une interface de dessin vectoriel. L'utilisateur peut : dessiner des formes (cercle, rond, ligne, carré, rectangle, triangle..) modifier leur couleur, leur taille de bordure, supprimer une forme, enregistrer un dessin et ouvrir un dessin déjà réaliser et enregistrer. Nous avons eu un délais de plus d'1 mois pour réaliser ce projet en binôme, nous avons pu également poser nos questions aux chargés de TP lors des séances.



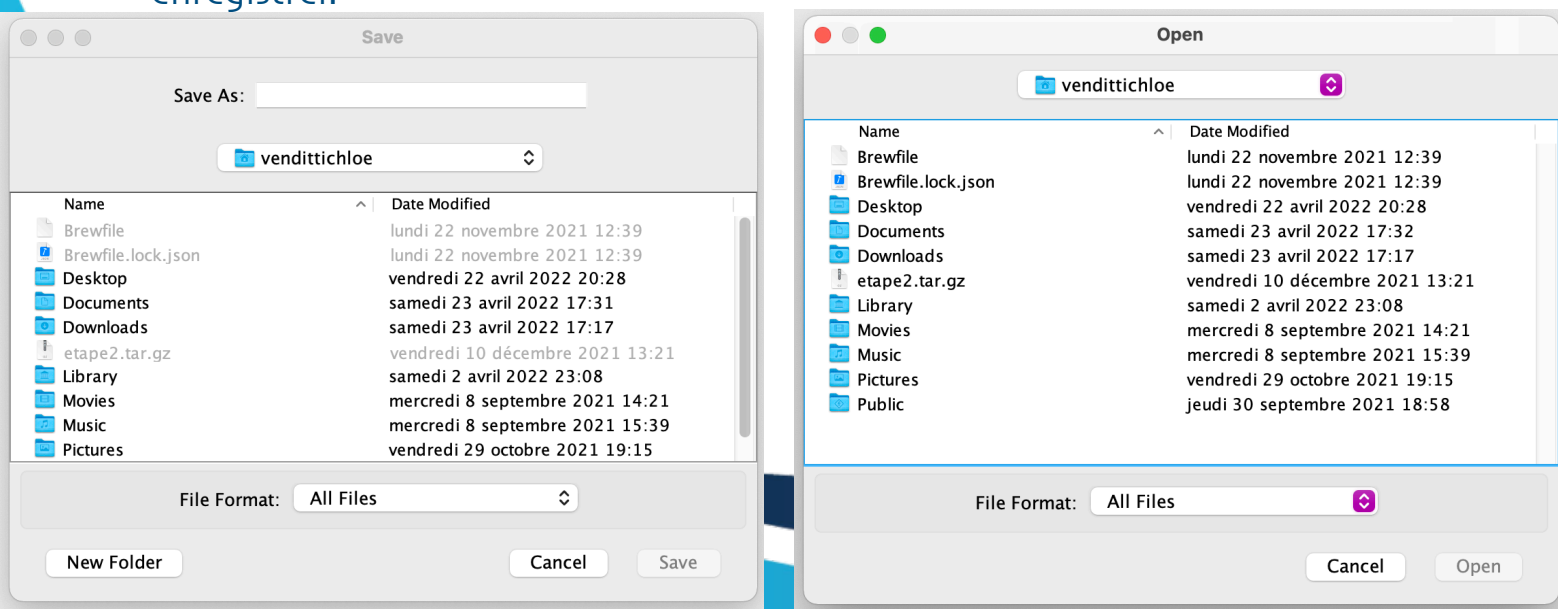
Étapes de la réalisation du projet :

Mise en place de l'interface générale :

La première étape a été de mettre en place l'interface générale et les outils (seulement le visuel). Nous avons créé notre JFrame dans notre code principal Main, le début de notre code principal (Main.java) importe des classes qui existent déjà dans des packages grâce à l'instruction **import** qui permet de se référer à des catégories qui sont déclarées dans d'autres paquets pour être accessible, notre JFrame accueillera toute l'interface. Un JMenu avec les outils : Fichier (ouvrir un dessin, enregistrer, nouveau dessin), Aide (si un soucis est rencontré). Puis deux JToolBar, une en haut et une à droite. Celle du haut permet à l'utilisateur de choisir quel dessin il souhaite réaliser, Formes ou Dessiner (créer des formes ou dessiner au stylo). Puis, celle de droite permet les modifications, Couleurs, Épaisseurs. Nous avons choisi de colorier l'interface en bleue, boutons et logos.

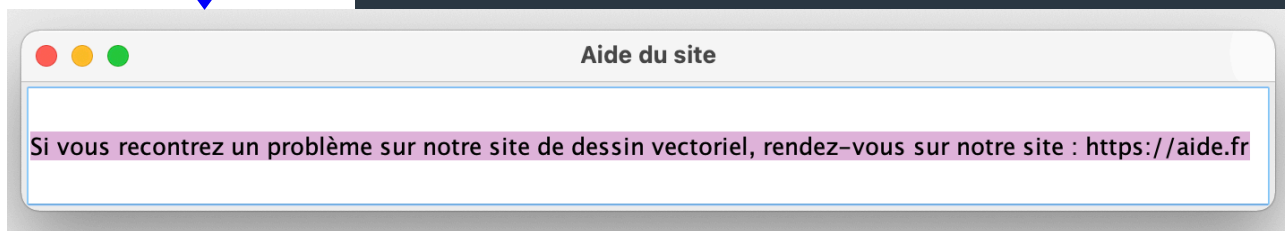
Mise en place des actions sur les boutons :

Ensuite, nous avons, pour chaque bouton, placé une action quand ils seront appuyés par l'utilisateur pour lui permettre d'utiliser l'interface. Le bouton Enregistrer le dessin permet à l'utilisateur d'enregistrer le dessin en cours. Le bouton Ouvrir un dessin permet à l'utilisateur d'ouvrir un fichier déjà enregistré.



Le bouton Nouveau dessin (en dessous de Fichier), permet à l'utilisateur d'obtenir une fenêtre « neuve » pour pouvoir dessiner. Pour cela nous avons utilisé : **`newf.addActionListener(e -> {`** pour ajouter au bouton correspondant une action, **`frame.dispose();`** est la première action réaliser lorsque le bouton est cliqué, la frame ouverte est fermée puis, **`frame.setVisible(true);`** on en ouvre une nouvelle vide. Lorsque le bouton Besoin d'aide (en dessous d'Aide) un petit message s'affiche : Lorsque le bouton

```
//Quand Aide est cliqué, un message d'aide apparaît
aide.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JFrame frame3=new JFrame("Aide du site");
        frame3.setBounds(200,200,700,100);
        frame3.setVisible(true);
        JTextField t1;
        t1=new JTextField("Si vous recontrez un problème sur notre site de dessin vectoriel, rendez-vous sur notre site : https://aide.fr");
        frame3.add(t1);
    }
});
```

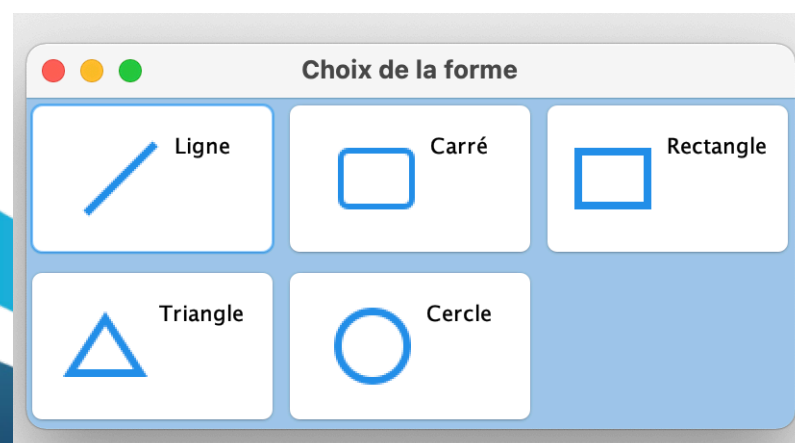


On ajoute une action au bouton Formes, une frame (frame2) s'ouvre puis on place un JPanel dedans avec un bouton pour chaque formes et un icône pour faciliter la lecture (ici ligne).

```
//on ajoute un événement au bouton, lorsqu'il est cliqué une nouvelle fenêtre apparaît pour le choix de forme du dessin
btn2.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent ae)
    {
        JFrame frame2=new JFrame("Choix de la forme");
        frame2.setBounds(400,300,400,200);

        //je crée un Panel pour le choix de formes du dessin avec des boutons pour chaque choix
        Panel c = new Panel();
        c.setLayout(new GridLayout(2,4,3,3));
        c.setBackground(new Color(167,199,231));

        //boutons pour chaque choix
        Icon icon_ligne=new ImageIcon("line-icon.png");
        JButton ligne = new JButton("Ligne", icon_ligne);
        ligne.setVerticalTextPosition(SwingConstants.TOP);
        c.add(ligne);
```



Le bouton Dessiner permet de dessiner directement avec la souris :

On permet à l'utilisateur de choisir la couleur du pinceau pour dessiner avant.

On a dédié une class pour la fenêtre Dessin au stylo.

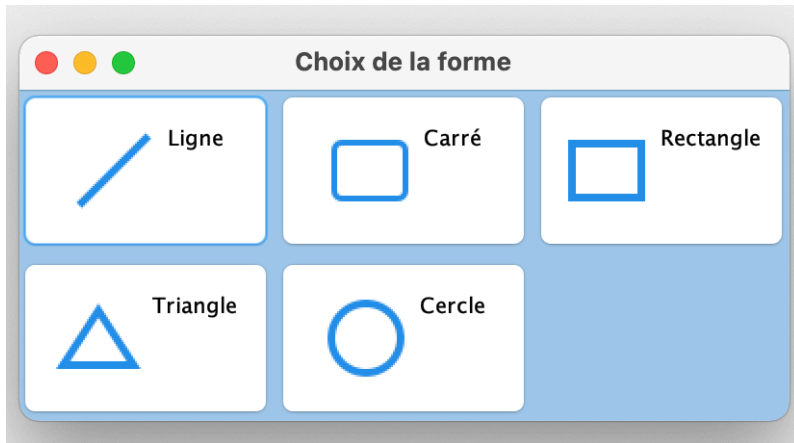
La classe dessiner correspond au traitement d'un événement de mouvement de souris, elle implémente donc l'interface d'écoute `MouseEvent`. Cette classe crée une frame qui va permettre à l'utilisateur de dessiner dessus.

Les 2 méthodes sont :

MouseDragged: Appelé lorsqu'un composant est enfoncé sur un composant. Dans cette méthode, un objet `Graphics` est construit qui va remplir un ovale délimité par le rectangle spécifié.

MouseMoved: Appelé lorsque le curseur de la souris a été déplacé sur un composant mais qu'aucun bouton n'a été enfoncé.





Lorsque l'utilisateur arrive sur cette fenêtre, il peut choisir quelle forme dessiner, dès qu'une forme est choisie une fenêtre de choix de couleurs apparaît, puis deux fenêtres permettant de définir les paramètres de taille (largeur, hauteur) de la forme. Puis on appelle la

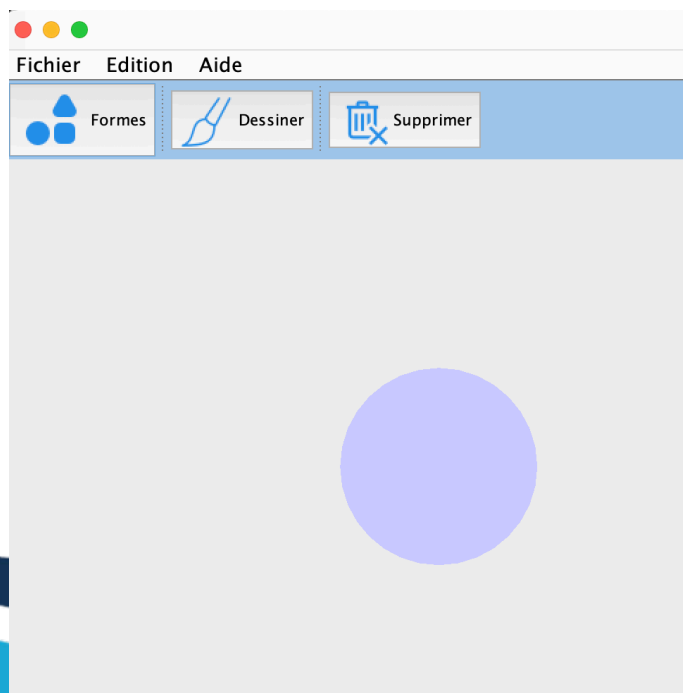
class de la forme sélectionnée pour la dessiner. Lorsque la forme est dessinée un bouton Supprimer est également affiché dans la toolbar du haut.

```
Icon icon_cercle=new ImageIcon("circle.png");
JButton cercle = new JButton("Cercle", icon_cercle);
cercle.setVerticalTextPosition(SwingConstants.TOP);
c.add(cercle);

//Dès que le bouton ligne est appuyé une ligne est dessinée
cercle.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent aae)
    {
        frame.getContentPane().add(new DrawMyCircle());
        toolbar_haut.add(supprimer);
    }
});
```

Ici, on prend l'exemple du cercle, on définit le bouton, puis on ajoute une action,

frame.getContentPane().add(new DrawMyCircle()); appelle la class DrawMyCircle qui dessine, déplace un cercle. Puis on ajoute le bouton Supprimer à la toolbar du haut.



Mise en place du dessin et des paramètres des formes :

Pour la fin, il nous fallait paramétrer les formes et les dessiner.
 Nous avons créer une class pour chaque forme; DrawMyCircle,
 DrawMyRect...
 Dans chaque class on :

- Paramètre le déplacement de la forme grâce à des mouseMotionListener :

```
//permet le déplacement de la forme avec la souris
private int mX,mY;

public DrawMyCircle(){
    addMouseListener(this);
    setVisible(true);
}

public void mouseMoved(MouseEvent me){
}

public void mouseDragged(MouseEvent me){
    mouseRelease(me);
}

public void mousePressed(MouseEvent me){
    this.repaint();
}

public void mouseRelease(MouseEvent me){
    this.mX = (int) me.getPoint().getX();
    this.mY = (int) me.getPoint().getY();
    this.repaint();
}
```

Quand la souris est cliqué on copie la forme dessiner, quand elle est bougé on peint la forme et réinitialise chaque position en récupérant la position de la souris actuelle puis, quand elle est relâchée, on peint la forme à la position de la souris actuelle.

- Paramétrer la forme (couleur + tailles) :

```
//permet à l'utilisateur de modifier la couleur
JColorChooser colorChooser = new JColorChooser();
Color couleur = JColorChooser.showDialog(null, "Choix de couleur ", Color.RED);
//permet à l'utilisateur de modifier les paramètres de taille de la forme

//épaisseur
JOptionPane jop = new JOptionPane();
//je convertie le String du JOptionPane initial en un int pour nos valeurs
int epaisseur = Integer.parseInt(jop.showInputDialog(null, "Veuillez choisir une épaisseur", "Épaisseur", JOptionPane.QUESTION_MESSAGE));

//hauteur
JOptionPane jop2 = new JOptionPane();
//je convertie le String du JOptionPane initial en un int pour nos valeurs
int hauteur = Integer.parseInt(jop2.showInputDialog(null, "Veuillez choisir une hauteur", "Hauteur", JOptionPane.QUESTION_MESSAGE));
```

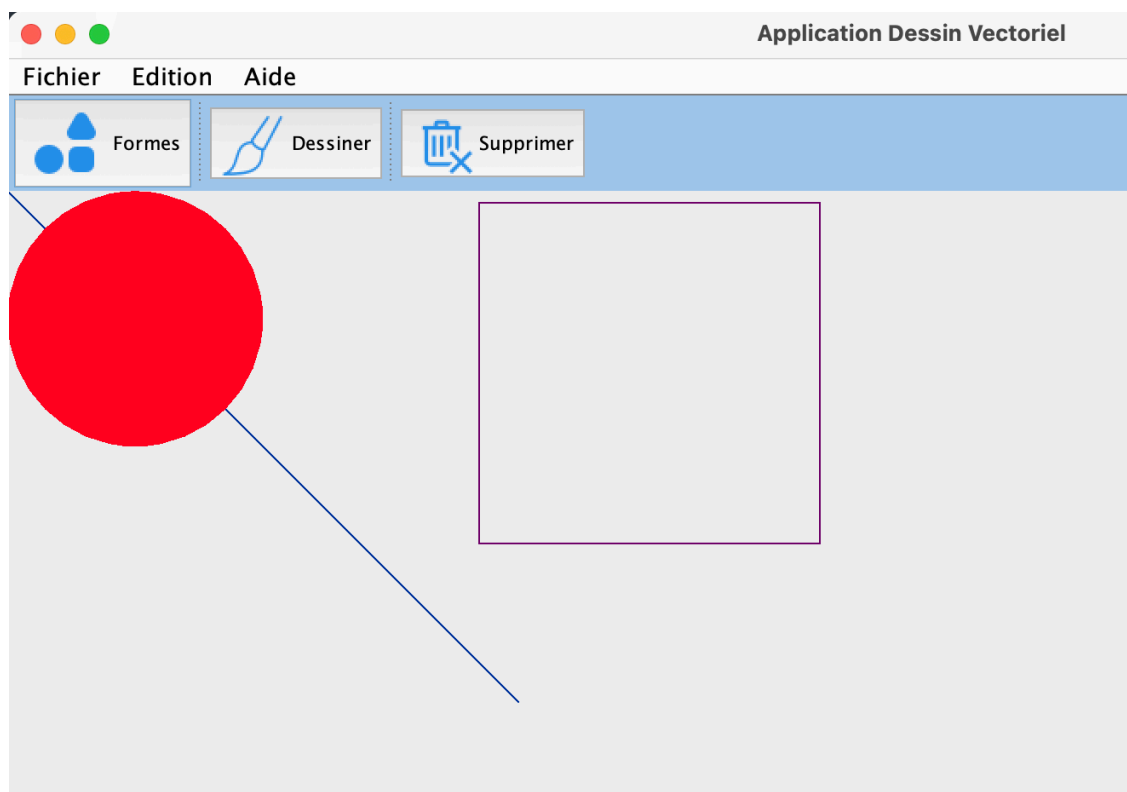

On choisi la couleur (couleur) JColorChooser, la largeur (épaisseur) JOptionPane puis la hauteur (hauteur) JOptionPane.

- Dessiner la forme avec les paramètres choisis par l'utilisateur :

```
public void paint(Graphics g){  
    g.setColor(couleur);  
    g.fillOval(mX, mY, epaisseur, hauteur);  
}
```

Exemple avec le cercle plein.

Exemple de résultat possible :



Problèmes rencontrés lors de la réalisation :

Problème avec l'interface :

Nous avons rencontrés plusieurs problèmes lors de la création de l'interface. Le premier a été avec les toolbar, nous en avons deux et le problème a été de les placer, une en haut et une verticale à droite.

```
JToolBar toolbar_haut = new JToolBar(JToolBar.HORIZONTAL);  
toolbar_haut.setFloatable(false);  
toolbar_haut.setRollover(true);  
toolbar_haut.setBackground(new Color(167,199,231));
```

La bordure EAST (droit) ne marchait pas, on a donc rajouté VERTICAL pour positionner notre toolbar droite, ce qui a résolu le problème.

```
//création de la JToolBar du côté droit  
JToolBar toolbar_droit = new JToolBar(JToolBar.VERTICAL);  
toolbar_droit.setFloatable(false);  
toolbar_droit.setRollover(true);  
toolbar_droit.add(btn);  
toolbar_droit.addSeparator();  
toolbar_droit.add(bord);  
frame.add(toolbar_droit, BorderLayout.EAST);
```

Problème de récupération de valeur :

Un des problèmes que nous n'avons pas su complètement résoudre a été de récupérer certaines valeurs pour les utiliser. Dans l'interface nous avons deux boutons à droite, Couleurs et Épaisseurs qui avaient pour but de modifier la couleur de la forme et modifier la taille de la forme. Le problème a été de récupérer la couleur choisie et la taille entrer par l'utilisateur.

```
//on crée un événement sur le bouton Couleur, dès qu'il est cliqué on affiche le panel de choix de couleurs
btn.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent event)
    {
        JColorChooser colorChooser = new JColorChooser();
        Color color = JColorChooser.showDialog(null, "Choix de couleur ", Color.RED);
        Color couleur = colorChooser.getColor();
    }
});
```

Ici, on récupère la couleur choisie dans « couleur »,

mais lorsque l'on dessine la forme le problème suivant apparaît :

```
projetoPOO — -zsh — 80x24
[vendittichloe@MacBook-Air-de-Venditti projetP00 % javac Main.java
./DrawMyCircle.java:65: error: cannot find symbol
    g.setColor(couleur);
               ^
    symbol:   variable couleur
    location: class DrawMyCircle
1 error
vendittichloe@MacBook-Air-de-Venditti projetP00 %
```

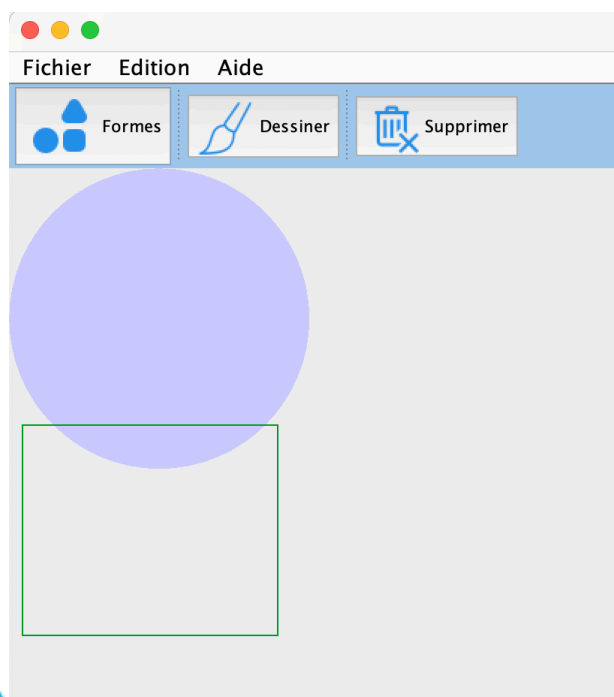
Pour régler le problème, dès que l'utilisateur clique sur une forme à dessiner, on lui propose de choisir la couleur qu'il souhaite. S'il ne souhaite pas de couleur, elle sera initialement noire.

Le même problème est rencontré pour l'épaisseur. On demande donc directement à l'utilisateur lorsqu'il choisit une forme de rentrer les tailles voulues.

Problèmes lors de l'affichage :

Nous avons quelques soucis lors de l'affichage des formes, effectivement, lorsque l'on dessine une seule forme, on peut la déplacer comme l'on veut. Si on dessine une deuxième forme, et qu'on souhaite la déplacer ce sera la première forme qui se déplacera et la deuxième forme disparaîtra de la frame, pour qu'elle se ré affiche il faut agrandir ou rétrécir un peu la fenêtre. Nous n'avons pas réussi à régler ce problème, puisqu'on utilise les mêmes méthodes pour déplacer les formes nous avons essayer de modifier le nom des valeurs et des événements pour rendre unique les méthodes mais cela n'a pas fonctionné.

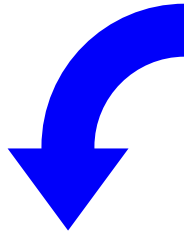
Néanmoins on peut dessiner plusieurs formes et en déplacer qu'une seule, mais chaque forme peut-être déplacée (si elle est la première forme dessiner avec d'autres OU si elle est la seule forme dessiner).



Ici, seul le carré a été bougé car le carré a été la première forme dessiner.

Mais, le rond peut être bouger si c'est la 1ère forme dessinée ou si elle est la seule forme dessinée.

Egalement un problème a été rencontré lors de la suppression d'une forme. Nous avons réussi à supprimer une forme mais le problème a été d'ajouter l'évènement a notre bouton supprimer qui se trouve dans notre code Main. L'erreur est la suivante :



```
//creation du bouton Supprimer
Icon icon_supp=new ImageIcon("supprimer.png");
JButton supprimer = new JButton ("Supprimer", icon_supp);

supprimer.setHorizontalTextPosition(SwingConstants.RIGHT);

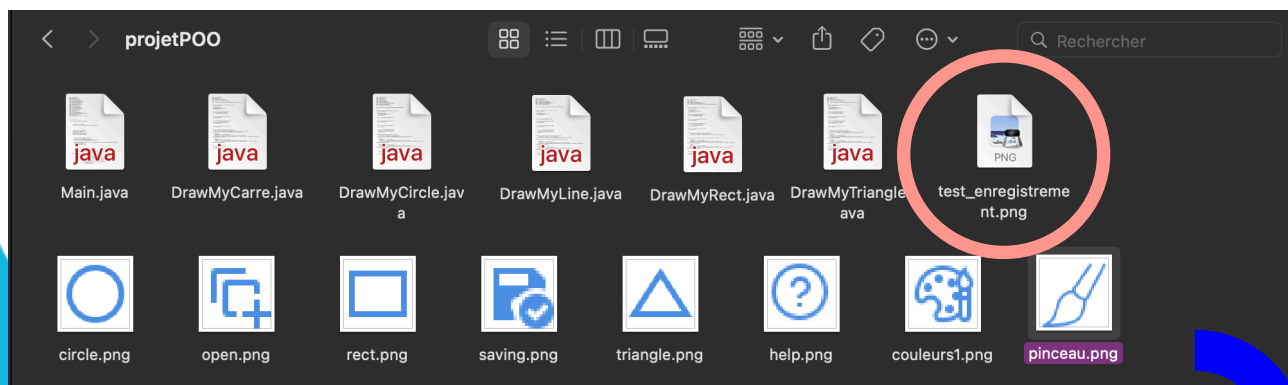
//supprimer l'élément quand le bouton est cliqué
supprimer.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent supp){

        supprimer.addActionListener(e -> g.clearRect(0, 0, 500, 500));

    }
});
```

```
vendittichloe@MacBook-Air-de-Venditti projetP00 % javac Main.java
Main.java:199: error: cannot find symbol
    supprimer.addActionListener(e -> g.clearRect(0, 0, 500, 500));
                                   ^
  symbol: variable g
1 error
vendittichloe@MacBook-Air-de-Venditti projetP00 %
```

Lors de l'enregistrement un dernier soucis est rencontré, le fichier sous l'extension .png s'enregistre bien mais il s'enregistre vide, il est donc pas possible de le visualiser.



Résultat final du projet et comment utiliser notre interface :

Une fenêtre intitulée « Application Dessin Vectoriel » va s'ouvrir.

En haut de la fenêtre se trouve un menu horizontal comprenant :

- Fichier : correspond à un menu déroulant avec 3 éléments : « Nouveau dessin » permettant l'ouverture d'une nouvelle page de dessin, puis « Ouvrir un dessin » pour chercher un dessin existant, et enfin « Enregistrer le dessin » pour la sauvegarde d'un travail réalisé.
- Edition.
- Aide : implémente un bouton aide qui renvoie sur notre page d'aide.

Notre fenêtre contient 2 barres d'outils horizontale et verticale sur le côté droit.

La barre d'outils horizontale implémente un bouton Formes et un bouton Dessin :

- Bouton "Formes" : propose 4 formes : Ligne, Carré, Rectangle, Triangle, Cercle.
- Bouton "Dessin" : permet à l'utilisateur de dessiner librement les formes de son choix.



La barre d'outils verticale implémente un bouton Épaisseur et Couleur :

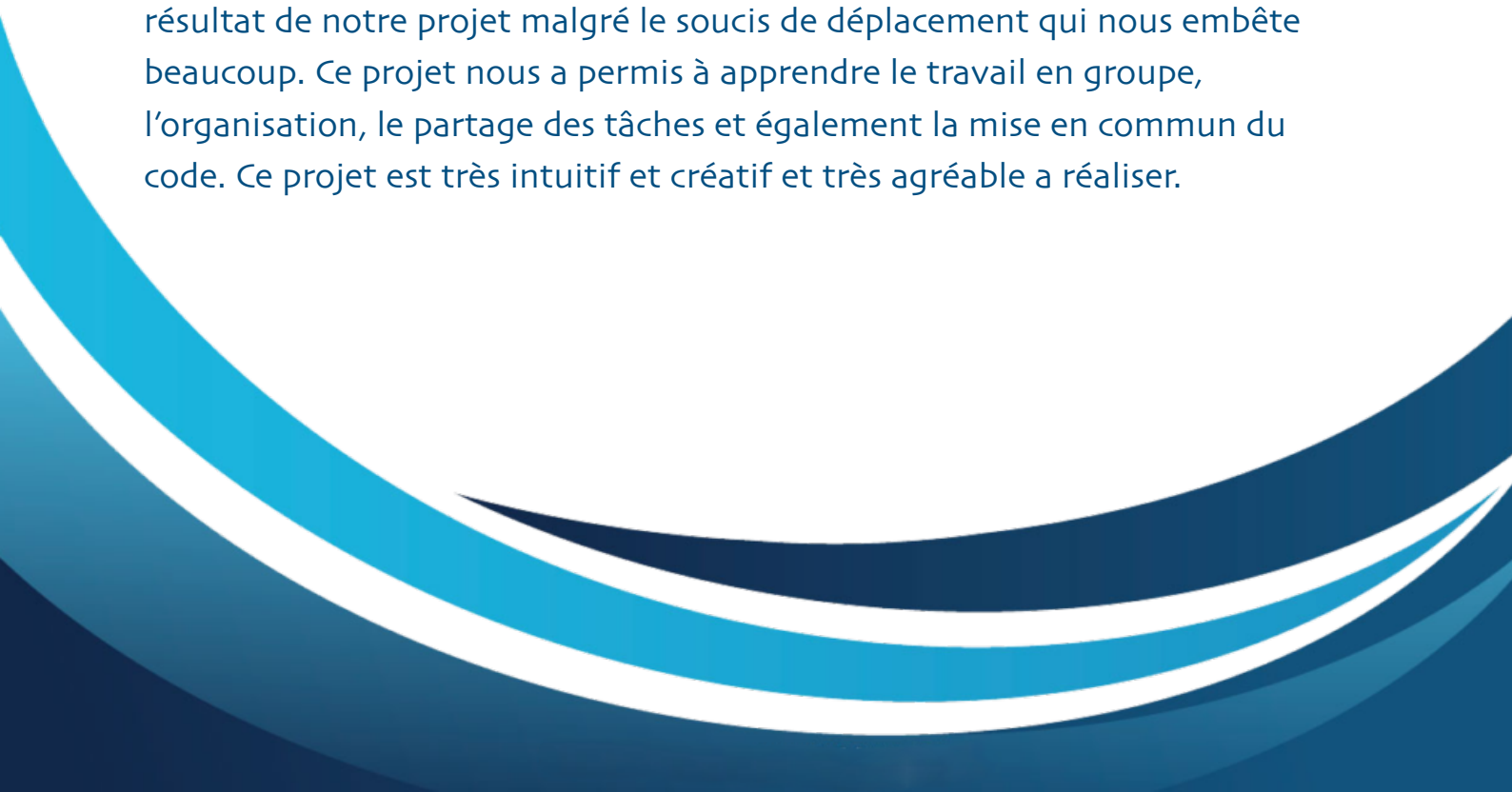
- Bouton "Couleurs" : présente un large panel de couleurs.
- Bouton "Épaisseur" : donne la possibilité à l'utilisateur de définir la hauteur et largeur de la forme dessinée.

(Ces deux boutons n'ont pas d'impact direct sur les formes dessiner, nous avons expliquer pourquoi dans la partie Problèmes rencontrés lors de la réalisation du projet).

Lorsque l'utilisateur a dessiner les formes qu'il souhaité, il peut supprimer la page avec le bouton Supprimer qui apparaît lorsqu'une forme est dessinée.

Conclusion :

Nous avons réussi à résoudre beaucoup de problèmes rencontrés pour avoir le meilleur résultat possible, qui rassemble ce qui a été demandé dans l'énoncé du projet donné. Malgré certains soucis, notamment lors du déplacement lorsque plusieurs formes sont dessiner et lors de l'enregistrement, nous avons réussis à dessiner, avec une couleur voulue et des tailles voulues, des formes. Nous avons également donné la possibilité à l'utilisateur de dessiner au stylo, pour faire un brouillon de ce qu'il veut réaliser avec les formes après par exemple. Nous sommes plutôt satisfaites du résultat de notre projet malgré le soucis de déplacement qui nous embête beaucoup. Ce projet nous a permis à apprendre le travail en groupe, l'organisation, le partage des tâches et également la mise en commun du code. Ce projet est très intuitif et créatif et très agréable a réaliser.



Bibliographie :

Voici une liste de liens qui nous ont aidés à la réalisation de notre projet :

- <https://duj.developpez.com/tutoriels/java/dessin/intro/#LIII-E>
 - <https://waytolearnx.com/category/programmation-java/interfaces-graphiques>
 - <https://fr.acervolima.com/mouselistener-et-mousemotionlistener-en-java/>
 - <https://askcodez.com/comment-puis-je-actualiser-ou-recharger-le-jframe.html>
 - <https://www.daniweb.com/programming/software-development/threads/422183/removing-all-components-from-jframe-application-hangs>
 - <https://www.delftstack.com/fr/howto/java/java-draw-triangle/>
 - <https://waytolearnx.com/2020/05/comment-tracer-des-lignes-rectangles-et-cercles-dans-jframe.html>
 - javadoc ([oracle.com](https://docs.oracle.com/javase/8/docs/api/))
- 