



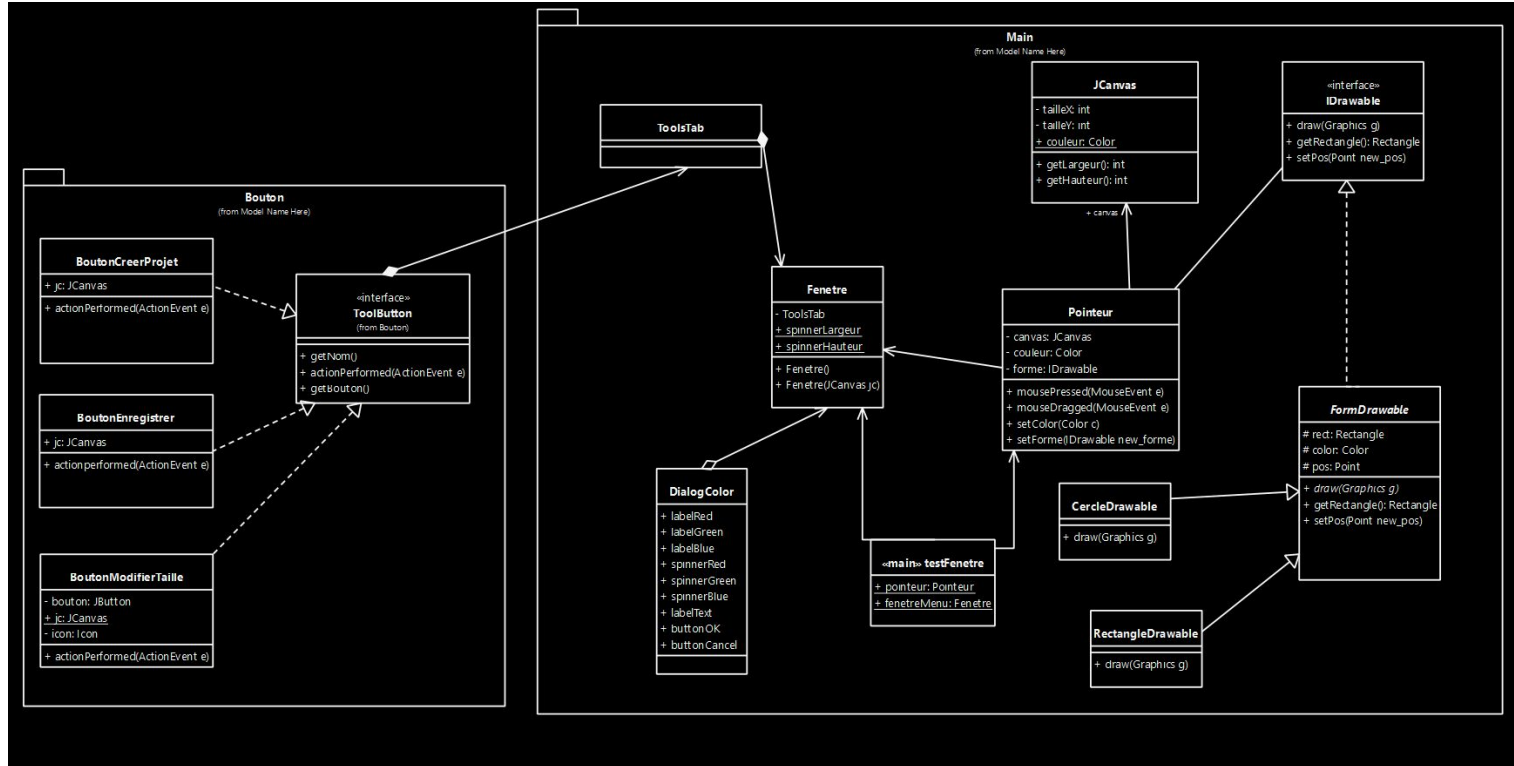
Projet long de technologie objet

Application de dessin : Draw7

Équipe 4 : ALBAREDE Nathan - BOSWELL Ethan - EL HOUR Othmane - MALAK Mathieu -
TRICHARD Matthieu - SCHLÖGEL Benjamin

Présentation technique

Diagramme UML



Les deux classes de base : JCanvas et Pointeur

```
public class Pointeur extends MouseInputAdapter {

    // On ajoute comme attribut du pointeur le canvas sur lequel il agit
    private JCanvas canvas;

    // Propriétés classiques du pointeur (taille, forme, couleur)
    private Color couleur;
    private IDrawable forme;
    private int taille;

    // Attributs gérant la mémoire (pour le Ctrl + Z)
    private Memory Memory;
    private Action act;

    public Pointeur(JCanvas c, Color couleur, Memory Memory) {
        super();
        canvas = c;
        taille = 8;
        forme = new CercleDrawable(this.couleur, new Dimension(taille, taille)); //forme par défaut
        canvas.addMouseListener(this);
        canvas.addMouseMotionListener(this);
        this.couleur = couleur;

        // Initialisation de la mémoire pour le pointeur (nécessaire pour le Ctrl + Z)
        this.Memory=Memory;
        Stack<Point> points= new Stack<Point>();
        Stack<Graphics> graphics= new Stack<Graphics>();
        this.act= new Action(points,graphics);
    }
```

```
@Override
public void mouseDragged(MouseEvent e) {
    Point p = e.getPoint();
    Graphics g=canvas.getGraphics();
    forme.setPos(p);
    forme.draw(g);
    this.act.getpoints().add(p);
    this.act.getgraphics().add(g);
}
```

Un exemple de bouton : le remplissage

```
/**Méthode qui remplit la surface de la couleur du pointeur
 *
 * @param x coordonnée de début de remplissage
 * @param y coordonnée de début de remplissage
 */
private void remplissageZone(int x, int y) {
    if (x < 0 || x >= canvas.getLargeur() || y < 0 || y >= canvas.getHauteur()) {
        return;
    }
    if (pixels[x][y] != null) {
        return; // Le pixel a déjà été rempli
    }

    Color currentColor = robot.getPixelColor(x, y);

    if (!currentColor.equals(firstColor)) {
        return; // Arrêter si la couleur du pixel ne correspond pas à la couleur courante
    }

    // Remplir le pixel
    testFenetre.pointeur.getForme().drawAuto(canvas.getGraphics(), testFenetre.pointeur.getColor(), x, y);

    // Appels récursifs : Les pixels adjacents afin de continuer le remplissage de
    // zone dans les quatre directions
    // remplissageZone(x - 1, y); // Pixel à gauche
    // remplissageZone(x + 1, y); // Pixel à droite
    // remplissageZone(x, y - 1); // Pixel au-dessus
    // remplissageZone(x, y + 1); // Pixel en dessous
}
```

```
public interface ToolButton {

    void actionPerformed(ActionEvent e);

    JButton getBouton();

    public String getNom();

}
```

```
// Créer des tâches pour les appels récursifs
Runnable leftTask = () -> remplissageZone(x - 1, y);
Runnable rightTask = () -> remplissageZone(x + 1, y);
Runnable aboveTask = () -> remplissageZone(x, y - 1);
Runnable belowTask = () -> remplissageZone(x, y + 1);

// Créer des threads pour exécuter les tâches en parallèle
Thread leftThread = new Thread(leftTask);
Thread rightThread = new Thread(rightTask);
Thread aboveThread = new Thread(aboveTask);
Thread belowThread = new Thread(belowTask);

// Démarrer les threads simultanément
leftThread.start();
rightThread.start();
aboveThread.start();
belowThread.start();

// Attendre la fin des threads
try {
    leftThread.join();
    rightThread.join();
    aboveThread.join();
    belowThread.join();
} catch (InterruptedException e) {
    e.printStackTrace();
}
```

Le retour arrière (Ctrl + Z)

```
public class Undoaction implements KeyListener {
    private Memory Memory;

    public Undoaction( Memory Memory) {
        this.Memory = Memory;
    }

    @Override
    public void keyPressed(KeyEvent e) {
        if (e.isControlDown() && e.getKeyCode() == KeyEvent.VK_Z) {
            if(!Memory.empty()) {
                Action act=Memory.pop();
                repaint(Memory,act);
            }
        }
    }
}
```

```
public void repaint(Memory memory, Action action) {
    IDrawable forme1 = action.getform();
    Color c=forme1.getcolor();
    forme1.setcolor(Color.white);
    Stack<Point> points1= action.getpoints();
    Stack<Graphics> graphics1= action.getgraphics();
    for(int i=0; i<points1.size();i++ ) {
        forme1.setPos(points1.get(i));
        forme1.draw(graphics1.get(i));
    }
    forme1.setcolor(c);
    for(Action act:memory) {
        IDrawable forme = act.getform();
        Stack<Point> points= act.getpoints();
        Stack<Graphics> graphics= act.getgraphics();
        for(int i=0; i<points.size();i++ ) {
            forme.setPos(points.get(i));
            forme.draw(graphics.get(i));
        }
    }
}
```

Enregistrement / Ouverture d'un projet

```
/** Méthode qui enregistre le dessin en faisant une capture d'écran de la fenêtre.
 *
 */
@Override
public void actionPerformed(ActionEvent e) {
    try {
        JCanvas canvas = BoutonCreerProjet.jc;

        Rectangle canvasRect = new Rectangle(canvas.getLocationOnScreen().x ,
            canvas.getLocationOnScreen().y ,
            canvas.getWidth(),
            canvas.getHeight());
        BufferedImage image = new Robot().createScreenCapture(canvasRect);

        // Enregistrer l'image dans un fichier PNG
        String source = JOptionPane.showInputDialog(null, "Veuillez indiquer le nom du fichier :");
        // Créer le nouveau fichier/image
        File file = new File(source + ".png");
        ImageIO.write(image, "png", file);

        System.out.println("Le dessin a été sauvegardé.");
    } catch (IOException | AWTException f) {
        f.printStackTrace();
    }
}
```

```
private JCanvas createCanvasWithImage(File file) {
    JCanvas canvas;
    try {
        BufferedImage image = ImageIO.read(file);
        imageWidth = image.getWidth();
        imageHeight = image.getHeight();
        canvas = new JCanvas(imageWidth, imageHeight);
        JLabel imageLabel = new JLabel(new ImageIcon(file.getAbsolutePath()));
        canvas.add(imageLabel);
        canvas.repaint();

    } catch (IOException e) {
        e.printStackTrace();
        canvas = null;
    }
    return canvas;
}

@Override
public void actionPerformed(ActionEvent e) {
    String source = JOptionPane.showInputDialog(null, "Entrez le chemin d'accès au fichier :");
    // Exemple : /home/mathieu/projet-long-alt/test.png
    // Pour spécifier le chemin d'accès au fichier que nous voulons ouvrir
    File file = new File(source);
    JCanvas canvas = createCanvasWithImage(file);
    pointeur = new Pointeur(canvas, Color.black, memory);
    fenetreDessin = new FenetreOuvrir(canvas, memory);

    fenetreDessin.setVisible(true);
}
```

Points à améliorer



- Pas de calques
- Remplissage beaucoup trop lent
- Boutons non fonctionnels pour une image ouverte
- Modifier Taille crée un nouveau JCanvas au lieu de modifier le JCanvas actuel
- Le Ctrl + Z ne fonctionne plus si on utilise un bouton

Méthodes agiles



Trello

Trello



Features

Tasks

User stories

Draw7 ☆ 📎 000 Tableau ▼

Power-up Calendrier Power-ups ⚡ Automatisation Filtre EB MM MT NA OH +1

Dessiner sur un tableau

Done

Tableau blanc, souris trace pixel couleur noir

🕒 28 mars - 31 mars 📄 2

📌 3/3

EB MM MT NA OH S

Done

Pointeur

🕒 22 mars - 26 mars 📄 3/3

EB MM MT NA OH S

+ Ajouter une carte

Interface graphique

Done

Changer taille de la feuille

🕒 3 avr. - 9 avr. 📄 1/1 MT S

Done

Barre d'outils

🕒 28 mars - 2 avr. 📄 2/2

EB MM MT NA OH S

Done

Menu principal

🕒 3 avr. - 9 avr. 📄 2/4 MT S

+ Ajouter une carte

Outils

Done

Changer couleur

🕒 14 avr. 🗨 1

Done

Gomme

🕒 14 avr. 📄 2/2

Loupe (zoom)

📄 0/1

En cours

Remplir (couleur)

🕒 21 avr. 📄 0/3 EB S

Insérer texte

📄 0/3

Done

Pipette

+ Ajouter une carte

Gestion de fichier

JPEG

PNG (transparence) ?

Done

Enregistrement du projet

🕒 4 mai

Done

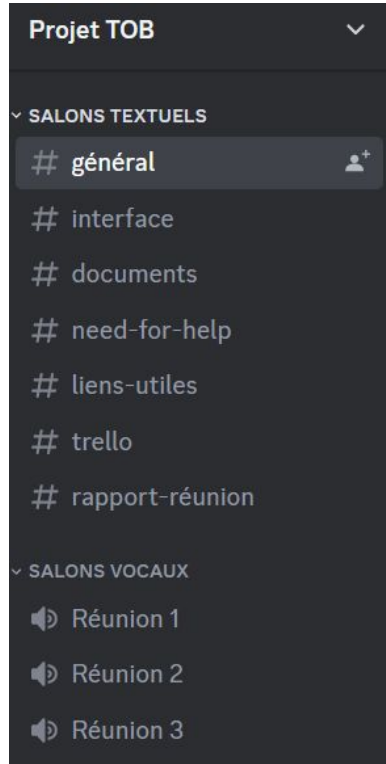
Ouverture d'un projet enregistré

🕒 11 mai

+ Ajouter une carte

Discord

- Différents salons en fonctions du sujet de conversation
- Une réunion par semaine (Dimanche à 17h)



Merci de votre attention !
