compte Rendu PIIA TP5

Boyuan ZHANG

# Q1

Dans la classe Paint1.java, la méthode utilisée pour associer les actions à la zone de dessin se reflète dans cette ligne de code :

***drawArea.setOnMouseClicked(e->mouseClick(e));***

Une expression lambda e->mouseClick(e) est utilisée ici pour définir le comportement lorsqu'un événement de clic de souris se produit. Cette expression signifie que lorsque l'on clique sur drawArea (canvas), la méthode mouseClick(MouseEvent e) est automatiquement appelée.

La méthode utilisée pour dessiner le cercle est gc.fillOval(double x, double y, double w, double h), qui est appelée sur l'objet GraphicsContext gc. Cette méthode dessine une ellipse remplie (un cercle dans ce contexte) dont la position et la taille sont déterminées par les paramètres de la méthode.

La méthode mouseClick(MouseEvent e) utilise l'événement de souris e pour déterminer la position du dessin du cercle et obtient les coordonnées du point de clic via e.getX() et e.getY(). Ces coordonnées sont ensuite utilisées comme gc Méthode .fillOval(...) Les paramètres x et y pour déterminer la position du cercle. La largeur et la hauteur du cercle sont codées en dur sur 10 pixels dans cet exemple.

# Q2 et Q3

Dans cette version, nous avons introduit la liste de tableaux de lignes pour stocker les coordonnées de début et de fin de chaque ligne. Dans la méthode handleMouseReleased, nous ajoutons les coordonnées de la ligne actuellement dessinée aux lignes. Dans la méthode handleMouseDragged, nous appelons d'abord la méthode draw pour effacer le canevas et redessiner toutes les lignes terminées, puis dessinons la ligne temporaire actuelle. De cette façon, chaque fois que la ligne temporaire change, ses traces précédentes seront effacées.

# 

# Q4

Pour class Line

*package TP5;*

*import javafx.scene.canvas.GraphicsContext;*

*// Cette classe représente un trait, c'est-à-dire une ligne définie par deux points.*

*public class Line {*

*double startX, startY, endX, endY; // Coordonnées de départ et d'arrivée du trait*

*// Constructeur pour créer un nouveau trait avec des coordonnées de départ et d'arrivée spécifiées*

*public Line(double startX, double startY, double endX, double endY) {*

*this.startX = startX; // Coordonnée X du point de départ*

*this.startY = startY; // Coordonnée Y du point de départ*

*this.endX = endX; // Coordonnée X du point d'arrivée*

*this.endY = endY; // Coordonnée Y du point d'arrivée*

*}*

*// Méthode pour dessiner le trait sur le canevas à l'aide du contexte graphique fourni*

*public void dessine(GraphicsContext gc) {*

*gc.strokeLine(startX, startY, endX, endY); // Dessine une ligne entre les points de départ et d'arrivée*

*}*

*}*

Ce code définit une classe `Line` pour représenter et dessiner une ligne définie par deux points dans un contexte graphique JavaFX. Les variables `startX`, `startY` représentent les coordonnées du point de départ, et `endX`, `endY` celles du point d'arrivée. Le constructeur `Line(double startX, double startY, double endX, double endY)` permet de créer un nouvel objet `Line` avec des coordonnées de départ et d'arrivée spécifiées. La méthode `dessine(GraphicsContext gc)` utilise le contexte graphique fourni (`GraphicsContext`) pour dessiner la ligne entre les points de départ et d'arrivée sur le canvas.