Résumé Main :

Le programme permet de créer et de manipuler une canevas en utilisant des commandes simples entrées par l'utilisateur. Le programme utilise un objet Canvas, qui est un tableau de pixels. Il contient des fonctions pour dessiner des formes, remplir des pixels, charger des images et afficher le canevas. Le programme boucle continuellement et attend une commande entrée par l'utilisateur, puis exécute cette commande et affiche le canevas mis à jour. Si l'utilisateur entre "exit", le programme se termine.

Les instructions pour lancer le programme :

Pour lancer le programme, vous devez suivre les étapes suivantes :

1. Installer Scala : Si vous n'avez pas déjà Scala installé sur votre ordinateur, vous pouvez le télécharger à partir du site officiel de Scala et suivre les instructions d'installation.
2. Compiler le code : Ouvrez votre terminal et accédez au répertoire où se trouve le fichier Main.scala. Pour compiler votre code, exécutez la commande suivante : scalac Main.scala

4. Exécuter le code : Une fois que le code a été compilé avec succès, vous pouvez l'exécuter à l'aide de la commande suivante : scala Main

La liste des différentes actions implémentées :

Le fichier Main contient les fonctions suivantes :
run() : permet de lancer la boucle principale de l'application en lisant une action sur la console, en l'exécutant et en affichant le résultat.

- execute(action: Seq[String], canvas: Canvas) : permet d'exécuter des actions selon la commande donnée. Elle renvoie le canvas mis à jour et l'état de la dernière exécution.
- display : permet d'afficher la canvas dans la console.
- update(pixel: Pixel) : permet de mettre à jour un pixel sur la canvas avec sa nouvelle couleur.
- updates(pixels: Seq[Pixel]) : permet de mettre à jour plusieurs pixels sur la canvas avec leur nouvelle couleur.
- update_color(pixel_to_modify: Pixel, newPixel:Pixel,firstColor:Char) : permet de mettre à jour la couleur d'un pixel spécifique sur la canvas.
- update_color1(pixel: Pixel, newPixel:Pixel) : permet de mettre à jour la couleur d'un pixel spécifique sur la canvas.
- updates_color(pixels: Seq[Pixel],newPixel : Pixel,firstColor: Char) : permet de mettre à jour la couleur de plusieurs pixels spécifiques sur la canvas.

Le programme contient aussi des fonctions appelées actions. Ci-dessous leurs noms et leurs rôles :

1. drawline(x1, y1, x2, y2, color): Cette fonction trace une ligne entre deux points (x1, y1) et (x2, y2) sur une canevas avec la couleur spécifiée. Cet algorithme est utilisé pour trouver les pixels qui se trouvent le long de la ligne entre les deux points et dessiner chacun de ces pixels en utilisant la couleur spécifiée.
2. drawline1(x1, y1, x2, y2, color, thickness)est similaire à drawline, mais elle permet également de spécifier l'épaisseur de la ligne en pixels.

3. load_image(filename): charge une image à partir d'un fichier et la retourne sous forme d'un tableau de pixels.

4. new_canvas(width, height) crée une nouvelle canevas avec la largeur et la hauteur spécifiées. Elle retourne une matrice de pixels avec des valeurs initialisées à zéro.
5. draw_fill_recursive(x, y, target_color, replacement_color, canvas) remplit une zone fermée à partir d'un point (x, y) avec la couleur de remplacement spécifiée (replacement_color). Elle utilise une approche récursive pour explorer les pixels adjacents à chaque pixel de la zone à remplir. Si le pixel a la même couleur que la couleur cible (target_color), il est remplacé par la nouvelle couleur (replacement_color).
6. update_pixel(x, y, color, canvas)met à jour la couleur d'un pixel aux coordonnées (x, y) sur un canevas avec la couleur spécifiée.
7. drawRectangle(x, y, width, height, color, thickness, canvas)dessine un rectangle sur une canevas avec la position et les dimensions spécifiées, ainsi que la couleur et l'épaisseur de la ligne. Elle utilise les fonctions drawline1 et update_pixel pour dessiner les lignes de bordure et remplir l'intérieur du rectangle avec la couleur spécifiée.