

DEMIR
Umut
2BTS SL

DM3 Interface Graphique Swing sous Java

Contexte:

Vous êtes stagiaire dans une entreprise, on vous demande de créer une application Java client lourd en SWING qui permet de remplir une liste déroulante par le nom des photos stockées dans un dossier. En sélectionnant chaque photo, cette dernière s'affiche dans la fenêtre. En cliquant sur le bouton Traiter, on pourra appliquer un traitement quelconque à l'ensemble des pixels de l'image.

Pour obtenir le résultat demandé, nous allons créer un nouveau projet, puis créer les fichiers **"PhotoFrame"** et **"PanelPhoto"**, et nous allons introduire le programme donné dans les consignes, en le modifiant, selon nos répertoires :

Le fichier **"PhotoFrame"**, contient tous les composants graphiques nécessaires à l'interface utilisateur. Permet à l'utilisateur de sélectionner une image depuis une liste déroulante (**JComboBox**) et de l'afficher. Ajoute un bouton ("Traiter") pour appliquer un traitement à l'image sélectionnée.

```
File folder = new
File("C:\\Users\\udemi\\eclipse-workspace\\DM3_Traitement_image\\src\\
tp7\\photos");
    if (!folder.exists()) {
        System.out.println("Le dossier n'existe pas : " +
folder.getAbsolutePath());
        return; // Arrête l'exécution si le dossier n'existe pas
    }
    // Liste des images avec les extensions .jpg et .png
    String[] images = folder.list((dir, name) ->
name.toLowerCase().endsWith(".jpg") ||
name.toLowerCase().endsWith(".png"));
    if (images == null || images.length == 0) {
```

```

        System.out.println("Aucune image trouvée dans le dossier !");
        return;
    }
    // Initialisation de la JComboBox avec les fichiers image
    jComboxPhotos = new JComboBox<>(images);
    // Chargement de la première image par défaut
    String nomPhoto = (String) jComboxPhotos.getSelectedItem();
    System.out.println("Image sélectionnée : " + nomPhoto);
    File imageFile = new File(folder, nomPhoto);
    BufferedImage bi = ImageIO.read(imageFile);
    pannelPhoto.setBi(bi);
    // Ajout des composants graphiques
    JPanel jPanelN = new JPanel();
    jPanelN.setLayout(new FlowLayout());
    jPanelN.add(jlbphotos);
    jPanelN.add(jComboxPhotos);
    jPanelN.add(jbtnTraiter);
    this.setBounds(10, 10, 800, 600);
    this.add(jPanelN, BorderLayout.NORTH);
    this.add(pannelPhoto, BorderLayout.CENTER);
    this.setVisible(true);
    // ActionListener pour JComboBox (changer l'image affichée)
    jComboxPhotos.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            try {
                String nomPhoto = (String)
jComboxPhotos.getSelectedItem();
                File imageFile = new File(folder, nomPhoto);
                if (imageFile.exists()) {
                    BufferedImage bi = ImageIO.read(imageFile);
                    pannelPhoto.setBi(bi);
                    pannelPhoto.repaint();
                } else {
                    System.out.println("Le fichier n'existe pas : " +
imageFile.getAbsolutePath());
                }
            }
        }
    });

```

```

        } catch (IOException e1) {
            e1.printStackTrace();
        }
    }
});
// ActionListener pour le bouton Traiter (appliquer un effet sur
l'image)
jbtnTraiter.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        BufferedImage bi = pannelPhoto.getBi();
        for (int i = 0; i < bi.getWidth(); i++) {
            for (int j = 0; j < bi.getHeight(); j++) {
                bi.setRGB(i, j, (bi.getRGB(i, j) * 5)); // Appliquer un
traitement simple
            }
        }
        pannelPhoto.repaint();
    }
});
} catch (IOException e) {
    e.printStackTrace();
}
}

```

Ensuite dans “**PhotoPanel**”, est une **classe personnalisée héritant de JPanel1**. Elle est conçue spécifiquement pour **afficher une image** à l'écran, voici son code :

```

package tp7;

import java.awt.image.BufferedImage;
import javax.swing.JPanel;
import java.awt.Graphics;
public class PannelPhoto extends JPanel {
    private BufferedImage bi;
    @Override

```

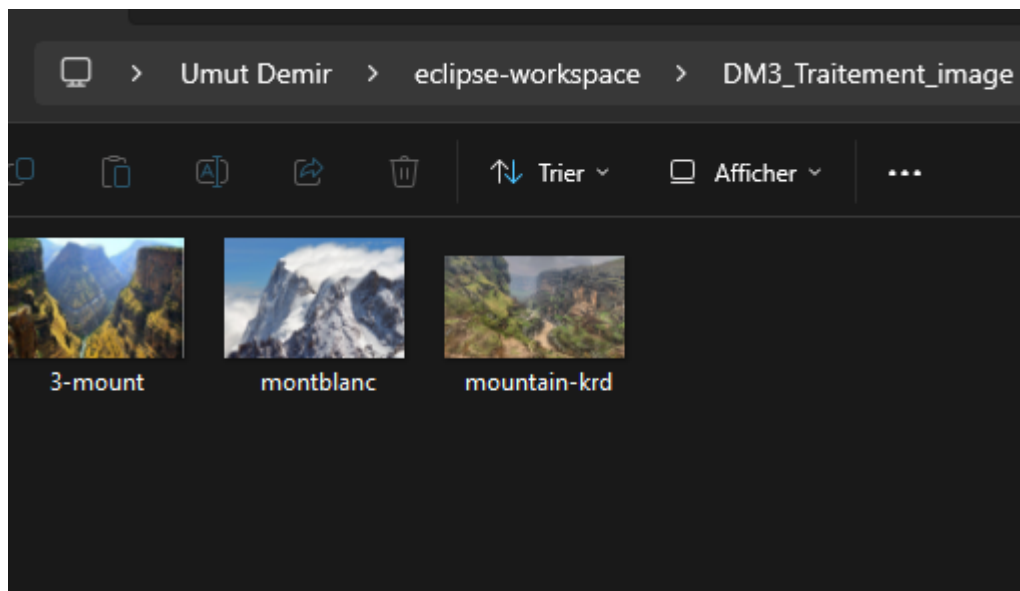
```

protected void paintComponent(Graphics g) {
    g.drawImage(bi, 0, 0, this.getWidth(), this.getHeight(),
        null);
}
public BufferedImage getBi() {
    return bi;
}
public void setBi(BufferedImage bi) {
    this.bi=bi;
}
}

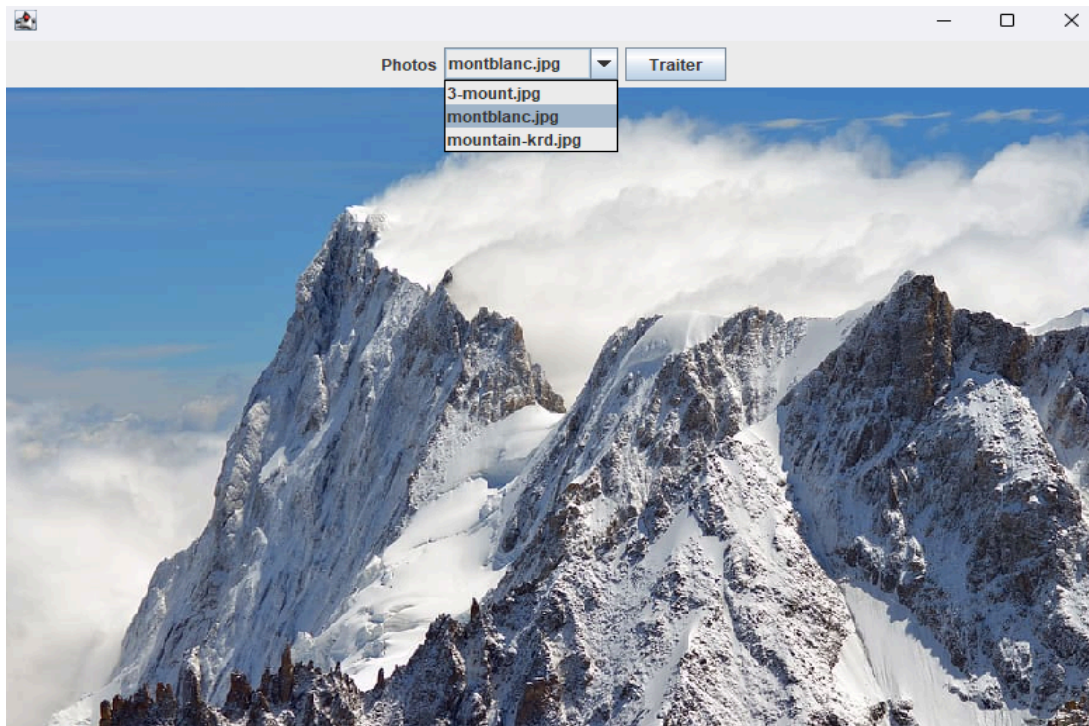
```

Pour finir voici le répertoire photo

“C:\Users\udem\workspace\DM3_Traitement_image\src\tp7\photos” :



Voici le résultat :



Lorsque j'appuie sur "Traiter", j'obtiens la pixellisation de l'image :

