import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.Image;

import java.awt.MediaTracker;

import java.awt.RenderingHints;

import java.awt.color.ColorSpace;

import java.awt.event.ActionEvent;

import java.awt.image.BufferedImage;

import java.awt.image.ColorConvertOp;

import java.awt.image.RenderedImage;

import java.io.File;

import java.io.IOException;

import java.net.URL;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

import javax.swing.JComboBox;

import javax.swing.JFileChooser;

import javax.swing.JPanel;

public class ImageProjet extends JPanel {

//Attributs

private Image image;

//Constructeurs

public ImageProjet(){

this.image = null; }

public ImageProjet(Image image){

this.image=image;

}

/\*public ImageProjet(String cheminImage){

this.image = getImage(getCodeBase(),cheminImage) ;

}

\*/

public ImageProjet(URL url){

BufferedImage img = null;

try {

img = ImageIO.read(url);

} catch (IOException e) {

}

this.image = img ;

}

//Methodes

public BufferedImage toBufferedImage() {

//source =http://java.developpez.com/faq/gui/?page=graphique\_general\_images

// On test si l'image n'est pas déja une instance de BufferedImage

if( image instanceof BufferedImage ) {

return( (BufferedImage)image );

} else {

//On s'assure que l'image est complètement chargée

image = new ImageIcon(image).getImage();

// On crée la nouvelle image

BufferedImage bufferedImage = new BufferedImage(

image.getWidth(null),

image.getHeight(null),

BufferedImage.TYPE\_INT\_RGB );

Graphics g = bufferedImage.createGraphics();

g.drawImage(image,0,0,null);

g.dispose();

return(bufferedImage);

}

}

public void nuanceDeGRis(){

//source =http://java.developpez.com/faq/gui/?page=graphique\_general\_images

BufferedImage bufferedImage = toBufferedImage();

ColorConvertOp op = new ColorConvertOp(

ColorSpace.getInstance(ColorSpace.CS\_GRAY),

null);

BufferedImage imageGrise = op.filter(bufferedImage,null);

image = imageGrise;

}

public void redimension(int width, int height) {

//source =http://java.developpez.com/faq/gui/?page=graphique\_general\_images

// On crée une nouvelle image aux bonnes dimensions.

BufferedImage buffer = new BufferedImage(width, height, BufferedImage.TYPE\_INT\_ARGB);

// On dessine sur le Graphics de l'image bufferisée.

Graphics2D g = buffer.createGraphics();

g.setRenderingHint(RenderingHints.KEY\_INTERPOLATION, RenderingHints.VALUE\_INTERPOLATION\_BILINEAR);

g.drawImage(image, 0, 0, width, height, null);

g.dispose();

// On retourne l'image bufferisée, qui est une image.

image = buffer;

}

/\*protected BufferedImage getImagePanneau(){

// récupérer une image du panneau

int width = this.getWidth();

int height = this.getHeight();

BufferedImage image = new BufferedImage(width, height, BufferedImage.TYPE\_INT\_RGB);

Graphics2D g = image.createGraphics();

this.paintAll(g);

g.dispose();

return image;

}\*/

/\*

protected void enregistrerImage ()throws IOException

{

BufferedImage bi = toBufferedImage();

int w = bi.getWidth(null);

int h = bi.getHeight(null);

if (bi.getType() != BufferedImage.TYPE\_INT\_RGB) {

BufferedImage bi2 = new BufferedImage(w, h, BufferedImage.TYPE\_INT\_RGB);

Graphics big = bi2.getGraphics();

big.drawImage(bi, 0, 0, null);

}

}

\*/

public void actionPerformed(ActionEvent e) {

JComboBox cb = (JComboBox)e.getSource();

if (cb.getActionCommand().equals("SetFilter")) {

int index = (cb.getSelectedIndex());

repaint();

} else if (cb.getActionCommand().equals("Formats")) {

/\* Save the filtered image in the selected format.

The selected item will be the name of the format to use

\*/

String format = (String)cb.getSelectedItem();

/\*Use the format name to initialise the file suffix.

Format names typically correspond to suffixes

\*/

File saveFile = new File("savedimage."+format);

JFileChooser chooser = new JFileChooser();

chooser.setSelectedFile(saveFile);

int rval = chooser.showSaveDialog(cb);

if (rval == JFileChooser.APPROVE\_OPTION) {

saveFile = chooser.getSelectedFile();

/\* Write the filtered image in the selected format,

\* to the file chosen by the user.

\*/

try {

ImageIO.write(toBufferedImage(), format, saveFile);

} catch (IOException ex) {

}

}

}

}

/\*

public void paint()

//http://www.commentcamarche.net/forum/affich-987970-affiche-une-image-en-java

//afficher l'image

{

Graphics g = toBufferedImage().createGraphics();

super.paintComponent(g);

//image = getToolkit().getImage("adresse de l'image"); Avoir l'image

if(image != null) // Si l'image existe, ...

g.drawImage(image, 200, 20, this); // ... on la dessine

}\*/

/\*

private static void downloadImage(String url, String imgSrc) throws IOException {

BufferedImage image = null;

try {

imgSrc = imgSrc.substring(imgSrc.lastIndexOf("/") + 1);

String imageFormat = null;

imageFormat = imgSrc.substring(imgSrc.lastIndexOf(".") + 1);

String imgPath = null;

imgPath = "C:/Users/Machine2/Desktop/CTE/Java-WebsiteRead/" + imgSrc + "";

URL imageUrl = new URL(url);

image = ImageIO.read(imageUrl);

if (image != null) {

File file = new File(imgPath);

ImageIO.write(image, imageFormat, file);

}

} catch (Exception ex) {

ex.printStackTrace();

}

}\*/

/\* public void enregistrerImage(){

RenderedImage rImage = (RenderedImage)(toBufferedImage());

File fichierArrive= new File("C:/Users/Public");

try {

ImageIO.write(rImage, "jpg", fichierArrive);

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

\*/

public Image getImage() {

return image;

}

public void setImage(Image image) {

this.image = image;

}

}//fin class

**2**

**private** **static** **void** **downloadImage**(String url, String imgSrc) **throws** IOException {  
 BufferedImage image = **null**;  
 **try** {  
 **if** (!(imgSrc.startsWith("http"))) {  
 url = url + imgSrc;  
 } **else** {  
 url = imgSrc;  
 }  
 imgSrc = imgSrc.substring(imgSrc.lastIndexOf("/") + **1**);  
 String imageFormat = **null**;  
 imageFormat = imgSrc.substring(imgSrc.lastIndexOf(".") + **1**);  
 String imgPath = **null**;  
 imgPath = "C:/Users/Machine2/Desktop/CTE/Java-WebsiteRead/" + imgSrc + "";  
 URL imageUrl = **new** URL(url);  
 image = ImageIO.read(imageUrl);  
 **if** (image != **null**) {  
 File file = **new** File(imgPath);  
 ImageIO.write(image, imageFormat, file);  
 }  
 } **catch** (Exception ex) {  
 ex.printStackTrace();  
 }  
  
 }

----------------------------------------------------------------------------------------------------------------------------

import java.awt.Graphics;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.IOException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.imageio.ImageIO;

/\*http://www.fobec.com/java/1078/convertir-une-image-nuance-gris.html

\* - ImageIO.read: charger l'image en couleur,

- BufferedImage: créer une image tampon grisée,

- drawImage: dessiner la photo sur le tampon,

- ImageIO.write: enregistrer l'image en noir et blanc.

\*/

/\*

\* Convertir une image couleur en nuances de gris

\*

\*/

public class ImageGrayScale {

//atributs

private static String inFilename ="imageEnCouleurs";

private static String outFilename ="imageEnNuancesDeGris";

File inputFile = new File(inFilename);

File outPutFile = new File(outFilename);

public ImageGrayScale(){

}

public void conversion(File inputFile,File outputFile) {

try {

System.out.println("Début de conversion....");

//Ouverture du fichier

BufferedImage imagesrc = ImageIO.read(inputFile);

//Convertion en grisé

BufferedImage imagedst = new BufferedImage(imagesrc.getWidth(),imagesrc.getHeight(), BufferedImage.TYPE\_BYTE\_GRAY);

//Enregistrer l'image au format PNG

File outFile = new File(outFilename);

ImageIO.write(imagedst, "PNG", outFile);

System.out.println("Fin de conversion....");

Graphics g = imagedst.getGraphics();

g.drawRect(0,0,100,100);

} catch (IOException ex) {

Logger.getLogger(ImageGrayScale.class.getName()).log(Level.SEVERE, null, ex);

}

}

//afficher l'image

public void afficherImage(BufferedImage imagesrc){

;

//g.drawImage(imagesrc, 0, 0, null);

//g.dispose();

}

}

public class Affichage extends Applet{

private Image bugatti=null;

@Override

public void paint(Graphics g) {

if(bugatti ==null) bugatti =getImage( this, "bugatti.png");

Graphics2D g2 =(Graphics2D)g;

g2.drawImage(bugatti, 0, 0, 250, 300, this);

super.paint(g);

}

public Image getImage(String path){

Image tempImage= null;

try{

URL imageURL = ImageLoadingTutorial.class.getResource(path);

tempImag = Toolkit.getDefaultToolkit().getimage(imageURL);

}

catch{Exception e){

System.out.println("An error "+e.getMessage());

}

return tempImage;

}