import java.awt.Color;

import java.util.Arrays;

class Collage {

public static void main(String[] args) {

Image canvas1 = ImageLibRef.canvasProvided(1280, 1280, new Color(0, 255, 0));

Image imageExample = CSE8ALib.readImage("image-files/pixel-heart.png");

int height = imageExample.getHeight();

int width = imageExample.getWidth();

Image imageExample1 = ImageLibRef.chromascale(imageExample, new Color(255, 0, 0));

Image imageExample2 = ImageLibRef.chromascale(imageExample, new Color(0, 255, 0));

Image imageExample3 = ImageLibRef.chromascale(imageExample, new Color(0, 0, 255));

Image Collage1 = ImageLibRef.overlayProvided(imageExample, canvas1, 0, 0);

Image Collage2 = ImageLibRef.overlayProvided(imageExample1, Collage1, 0, height);

Image Collage3 = ImageLibRef.overlayProvided(imageExample2, Collage2, width, 0);

Image Collage4 = ImageLibRef.overlayProvided(imageExample3, Collage3, width, height);

System.out.println("canvas1: " + canvas1.toString());

CSE8ALib.explore(Collage4);

}

}

ImageLib.java

import java.awt.Color;

import java.util.Arrays;

class ImageLib {

public static void main(String[] args) {

}

//1.scaleTo

public static Color scaleToColor(Image original, int targetWidth, int targetHeight, int col, int row) {

int originalCol = CSE8ALib.doubleToInt(col\*original.getWidth()/targetWidth);

int originalRow = CSE8ALib.doubleToInt(row\*original.getHeight()/targetHeight);

return original.getColorAtPos(originalCol, originalRow);

}

public static Color[] scalePixels(Image original, int targetWidth, int targetHeight) {

int size = targetWidth \* targetHeight;

Color[] newArray = new Color[size];

for(int col = 0; col < targetWidth; col += 1) {

for(int row = 0; row < targetHeight; row += 1) {

newArray[row \* targetWidth + col] = ImageLib.scaleToColor(original, targetWidth, targetHeight, col, row);

}

}

return newArray;

}

public static Image scaleTo(Image original, int targetWidth, int targetHeight) {

int width = targetWidth;

int height = targetHeight;

Color[] pixels = ImageLib.scalePixels(original, targetWidth, targetHeight);

return new Image(targetWidth, targetHeight, pixels);

}

2.// canvas

public static Color[] canvasPixels(int width, int height, Color color){

int size = width \* height;

Color[] newArray = new Color[size];

for(int i = 0; i < size; i += 1) {

newArray[i] = color;

}

return newArray;

}

public static Image canvas(int width, int height, Color color) {

Color[] pixels = ImageLib.canvasPixels(width, height, color);

return new Image(width, height, pixels);

}

//5. overlay

public static Color[] overlayPixels(Image fg, Image bg, int topLeftCol, int topLeftRow) {

int fgWidth = fg.getWidth();

int fgHeight = fg.getHeight();

int bgWidth = bg.getWidth();

int bgHeight = bg.getHeight();

int size = bgWidth \* bgHeight;

Color[] newArray = new Color[size];

for(int col = 0; col < bgWidth; col+= 1) {

for(int row = 0; row < bgHeight; row+=1 ){

if(col>=topLeftCol && col<(topLeftCol+fgWidth) && row>=topLeftRow && row<(topLeftRow+fgHeight)){

newArray[row\*bgWidth+col] = fg.getPixels()[(row-topLeftRow)\*fgWidth + (Col-topLeftCol)];

}

else {

newArray[row\*bgWidth + col] = bg.getPixels()[row\*bgWidth + col];

}

}

}

return newArray;

}

public static Image overlay(Image fg, Image bg, int topLeftCol, int topLeftRow){

int width = bg.getWidth();

int height = bg.getHeight();

Color[] pixels = ImageLib.overlayPixels(fg, bg, topLeftCol, topLeftRow);

return new Image(width, height, pixels);

}

}

PA7

class PA7 {

public static void main(String[] args) {

Color[] examplePixels = {new Color(255, 0, 0), new Color(0, 255, 0), new Color(0, 0, 255),

new Color(0, 0, 0), new Color(128, 128, 128), new Color(255, 255, 255)};

Image exampleImage = new Image(3, 2, examplePixels);

Image checker = CSE8ALib.readImage("image-files/checker5.bmp");

Image crane = CSE8ALib.readImage("image-files/crane.jpg");

//1.scaleTo

Image scaleTo1 = ImageLib.scaleTo(exampleImage, 3, 3);

Color[] scaleTo1PixelsExpect = {

new Color(255, 0, 0), new Color(0, 255, 0), new Color(0, 0, 255),

new Color(255, 0, 0), new Color(0, 255, 0), new Color(0, 0, 255),

new Color(0, 0, 0), new Color(128, 128, 128), new Color(255, 255, 255)};

System.out.println("scaleTo1 image : " + scaleTo1.toString());

System.out.println("scaleTo1 image expect: Image[width=3,height=3]");

System.out.println("scaleTo1 pixels: " + Arrays.toString(scaleTo1.getPixels()));

System.out.println("scaleTo1 pixels expect: " + Arrays.toString(scaleTo1PixelsExpect));

Image scaleTo2 = ImageLib.scaleTo(exampleImage, 2, 2);

Color[] scaleTo2PixelsExpect = {

new Color(255, 0, 0), new Color(0, 255, 0),

new Color(0, 0, 0), new Color(128, 128, 128),

};

System.out.println("scaleTo2 image : " + scaleTo2.toString());

System.out.println("scaleTo2 dims expect: Image[width=2,height=2]");

System.out.println("scaleTo2 pixels: " + Arrays.toString(scaleTo2.getPixels()));

System.out.println("scaleTo2 pixels expect: " + Arrays.toString(scaleTo2PixelsExpect));

Image scaleTocrane = ImageLib.scaleTo(crane, 200, 100);

CSE8ALib.explore(scaleTocrane);

// 2.canvas

Color canvasExample1 = new Color(0, 255, 0);

Image canvas1 = ImageLib.canvas(2, 2, canvasExample1);

Color[] canvas1PixelsExpect= {

new Color(0,255,0),new Color(0,255,0),

new Color(0,255,0),new Color(0,255,0)};

System.out.println("canvas1 image: " + canvas1.toString());

System.out.println("canvas1 image expect: Image[width=2, height=2]");

System.out.println("canvas1 pixels: " + Arrays.toString(canvas1.getPixels()));

System.out.println("canvas1 pixels expect: " + Arrays.toString(canvas1PixelsExpect));

Color canvasExample2 = new Color(255,0,0);

Image canvas2 = ImageLib.canvas(3,3,canvasExample2);

Color[] canvas2PixelsExpect={

new Color(255,0,0),new Color(255,0,0),new Color(255,0,0),

new Color(255,0,0),new Color(255,0,0),new Color(255,0,0)};

System.out.println("canvas2 image: " + canvas2.toString());

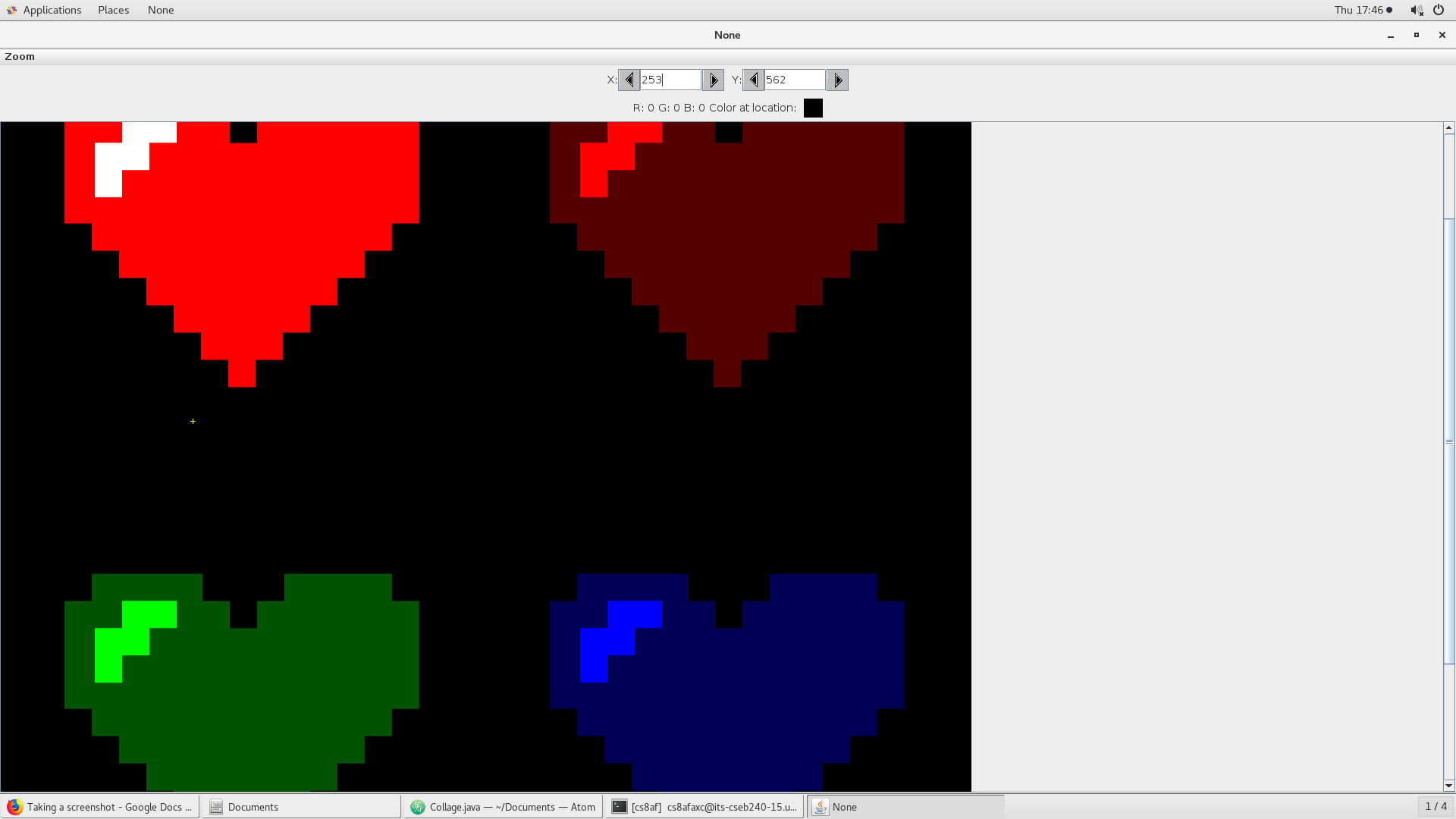
System.out.println("canvas2 image expect: Image[width=3, height=3]");

System.out.println("canvas2 pixels: " + Arrays.toString(canvas2.getPixels()));

System.out.println("canvas2 pixels expect: " + Arrays.toString(canvas2PixelsExpect));

Image canvasImage= ImageLib.canvas(200,100,canvasExample1);

CSE8ALib.explore(canvasImage);

 name the image as “color-collage.png” and save it in image-files