Design of user interfaces

Procesamiento de imagenes

2018/19 course - Practice 4

David Bohmann
Petr Lukašík

October 4, 2018

Contents

1. Assigment

In the previous practice we studied the basic primitives that Java provides for drawing. In this practice, another set of functions will be used to handle images (BufferedImage class) with the aim of introducing the RadioButton and JCheckBox classes, which are a particular case of buttons. The objective of this practice will therefore be to use the radio button component to select between three different images and with the check box component indicate which processing will be applied to the selected image.

The developed application will have the following functionality:

- By using three radio buttons it will allow the user to select what image will be processed.
- A smoothing and/or highlighting process can be applied to the image using a check box area that contains the two options.
- The image will be displayed each time it is selected and updated each time that the processing is applied to it.

1.1 Implementation notes

Image filtering is based on the convolution of the image with a certain mask but because it is not the subject of this lesson, the code is shown below to perform the filtrations that are requested in this practice.

Smoothing an image (BufferedImage)

```
float [] difuminar = {
    0.111f, 0.111f, 0.111f,
    0.111f, 0.111f, 0.111f,
    0.111f, 0.111f, 0.111f};

Kernel sharpkernel = new Kernel (3, 3, blur);
ConvolveOp sop = new ConvolveOp (sharpkernel, ConvolveOp.
   EDGE_NO_OP, null);
image = sop.filter (image, null)
```

Enhancement of an image (BufferedImage)

```
resaltar = {
    0.f, -1.f, 0.f,
    -1.f, 5.0f, -1.f,
    0.f, -1.f, 0.f};

Kernel sharpkernel = new Kernel (3, 3, highlight);
ConvolveOp sop = new ConvolveOp (sharpkernel, ConvolveOp.
   EDGE_NO_OP, null);
image = sop.filter (image, null)
```

2. Implementation

Our application (shown in pic 2.1) is based on assigment and contains two main areas. First area is image view on the center/left side. Second area is the main settings which is positioned on the right side of the application. The radio buttons (pic 2.2a) in the settings area are placed in vertical position. They allow to switch between three different pictures. Below radio buttons are checkboxes (pic 2.2b), which are upon checking applying highlight or blur filter on the current image. Every action either on radio buttons or checking the checkbox instantly refreshes image and applies every checked filter. The application is not resizable so we don't have to deal with image resizing problems.

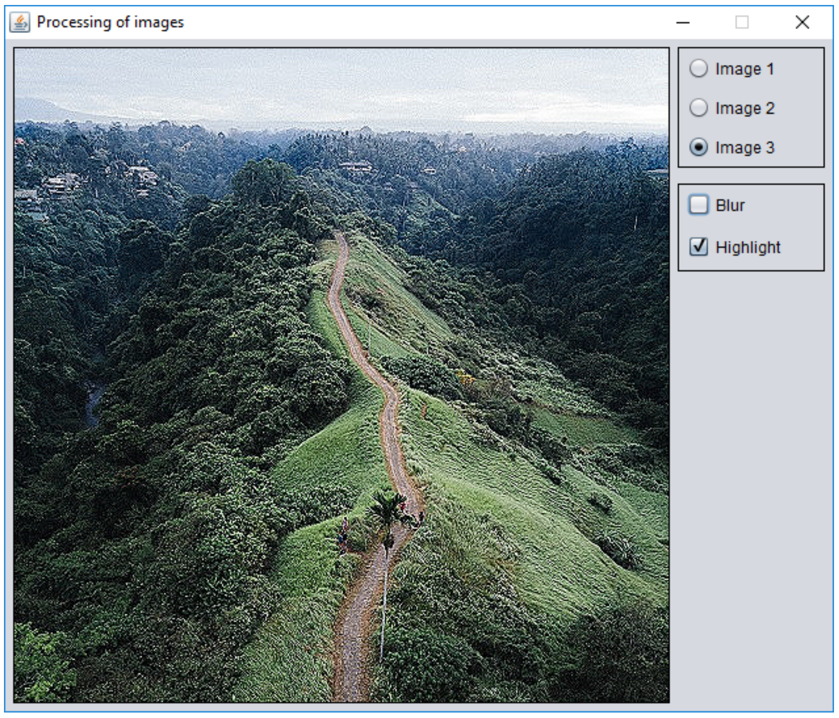
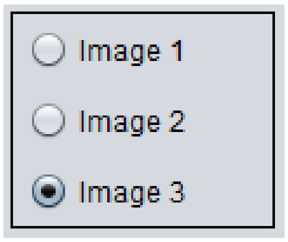
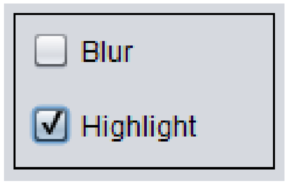


Figure 2.1: Image filtering application



(a) Detail on radio buttons



(b) Detail on checkboxes

Figure 2.2: Main settings elements

3. Shneiderman and Plaisant principles

- **Consistency**
  All main areas are having thin black border to show main area groups for the user. Left area is showing image with results, right area is showing settings.
- **Universal usability**
  We did not implement any shortcuts or hidden functions to ease the using of the application because there are not that many interactions for this to be neccessary.
- **Informative feedback**
  For every input done, image area changes. There is no more need to inform user what is happening.
- **Design dialogue to yield closure**
  Changing inputs displays results instantly in the image area (ref Informative feedback).
- **Simple error handling.**
  There are no choices to make an error. No error labels were implemented.
- **Permit easy reversal of actions**
  Every action done can be switched back to the previous state by checking unchecking right buttons.
- **Support internal locus of control**
  The user is in full control of choosing filters and predefined images. Every action is instantly applied and shows results. Of course, user could have more options like choosing his own image by filechooser.
- **Reduce short-term memory load**
  For such a small application our application is fast enough for human needs.