University of Puerto Rico
Department of Electrical and Computer Engineering

ICOM 4015 Laboratory: Advanced Programing

Laboratory 5: Image Transitions

Completed by:
ID:
Date:

1   Introduction

In this laboratory we will learn how to use iteration to copy images from one frame to another using different transition effects.

2   Contents:

This laboratory will make use of 3 classes:

1. LeftPanel
2. RightPanel
3. TransitionTester

Import into Eclipse the archive file provided along with this document to access the classes. If you have any doubts, the T.A. will teach you how to import compressed projects into the workspace.

3. Classes overview: Make sure you spend some time reading the code and understanding how these classes work before you come to the lab.

1. **LeftPanel :** This class represents a JPanel that holds a source image of your choice.

2. **RightPanel :** This class represents a JPanel that displays the same image displayed on the source (left) window repeatedly, each time using a different transition effect.

3. **TransitionTester:** This class displays a file chooser to select an image file. Then it creates two frames, displays the selected image once on the left frame, and goes into an infinite loop continuously redisplaying the image on the right frame each time using a different transition.

Run the application as you received it and study the code to understand how everything works. In particular figure out how the *repaint* method from the *JFrame* class is used in combination with delays in order to get the desired transition effects. Select an image of your choice that is wider than taller and no larger than 400 pixels wide by 500 pixels tall.

The following method implements a horizontal transition that displays pixels on each row from left to right. Rows are displayed from top to bottom.

```
//
// Transition by rows left to right top to bottom
public void transitionLRTB(Graphics g, BufferedImage leftImage) {
    int width = leftImage.getWidth();
    int height = leftImage.getHeight();
    for (int j=0; j<height; j++){
        for (int i=0; i<width; i++){
            int pixelColor= leftImage.getRGB(i,j);
            img.setRGB(i, j, pixelColor);
        }
        repaint();
        try { Thread.sleep(10); } catch (InterruptedException e) { };
    }
}
```

Lab Q1:
Complete the definition of the *TransitionTBLR* method in order to implement a vertical transition that displays pixels on each column from top to bottom and then each column from left to right. To test your method you may find useful to complete Q2 simultaneously. Paste the definition of your method in the area bellow delimited by the /'s.

//
//

Lab Q2:
Modify the *main method* in the *TransitionTester* class in order to incorporate the new transition to the sequence of transitions used to display the image on the right frame. Paste the modified version of the switch statement inside the main method in the area bellow delimited by /'s.

//
//

Lab Q3:
The method *transitionDiagonal45LR* implements a 45-degree diagonal transition from left to right. The method even works when images are wider than taller. However, when the image is taller than wider, the method does not work correctly. Try the program again with such an image and analyze its behavior. Correct the implementation of the method and paste your new version in the area below delimited by /'s.

//
//

Lab Q4:

Implement a new *transitionRandom* method in the *RightPanel* class that display the source image on the right frame by selecting one random pixel at a time. The transition should only attempt to copy enough pixels to make the image visible since copying all pixels may take a long time. To test your method you may find useful to complete Q5 simultaneously. Paste the code for your *transitionRandom* method in the area below delimited by /'s.

//
//

Lab Q5:

Modify the *main method* in the *TransitionTester* class in order to incorporate the new *transitionRandom* method to the sequence of transitions used to display the image on the right frame. Paste the modified version of the switch statement inside the main method in the area bellow delimited by /'s.

//
//