Tutorial 7: JPanel & JButton

by H.Darbandi

Part 1:

The aim of this tutorial is to learn image buttons and nested JPanel.
Note that our approach in this tutorial is just for learning purposes, and this is not a prescription for doing GUI in Java. Every application needs its own approach, and this is only one of them.

1. Download the tutorial_8.zip file. Run and study it. It creates an image button.
ImageButton extends JButton and draws an image on the button.
SwitchImage(), changes the button image to the next image in the array of image files.
MyPanel extends JPanel , and creates an ImageButton and attaches it to the MyPanel. Note that we override the paintComponent(...) of its super class (JComponent) that calls the paintComponent() of the super class first before doing anything.

```
public void paintComponent(Graphics g) {
    Graphics2D g2 = (Graphics2D) g;
    super.paintComponent(g);
    imgBtn.repaint();
}
```

2. Clicking the button has no effect, since we are not attached any listener to it yet.
First we need to create a class that implements interface ActionListener, and override it's actionPerformed method:

```
class ImageButtonInvoker implements ActionListener {
      public void actionPerformed(ActionEvent event) {
            // Handle action
      }
}
```

Then we add an action handler; for example invoke the switchImage( ) of the class ImageButton:

```
class ImageButtonInvoker implements ActionListener {
      public void actionPerformed(ActionEvent event) {
            imgBtn.switchImage();
      }
}
```

and finally we have to create an instance of the class ImageButtonInvoker and attach it to the button:

```
imgBtn.addActionListener(new ImageButtonInvoker());
```

Putting everything together, the createCrButton( ) is modified as shown bellow:

```
private void createCrButton() {
     imgBtn = new ImageButton();
     add(imgBtn);
     class ImageButtonInvoker implements ActionListener {
           public void actionPerformed(ActionEvent event) {
                 imgBtn.switchImage();
```

```
            }
      }
      imgBtn.addActionListener(new ImageButtonInvoker());
}
```

Now whenever you click the mouse, it invokes the switchImage() of the button, and the image of the button changes.

3. Now lets create multiple click-able buttons with the similar functionality:
Create class MainPanel that extends JPanel, and sets its layout to a GridLayout. Create more instances of MyPanel class and add them to the MainPanel. The constructor of the MainPanel class is shown below:

```
public class MainPanel extends JPanel {
      private final int SIZE = 3;
      public MainPanel()   {
            setLayout(new GridLayout(SIZE,SIZE));
            for(int i=0; i<SIZE*SIZE; i++)
                  add(new MyPanel());
      }
}
```

Now you have to add an instance of this class to your Jframe. Modify the main method of the Application class, and instead of adding MyPanel() to the frame, add an instance of the MainPanel to it.
Remove:

```
  MyPanel myPanel = new MyPanel();
  frame.add(myPanel);
```

and add

```
  MainPanel mainPanel = new MainPanel();
    frame.add(mainPanel);
```

If you do it right, you should have an application that each button changes its image by being clicked.

Part 2:

You can also modify the size of the image before attaching it to your button. Modify the setIcon(...) of ImageButton class as shown below:

```
private void setIcon(String fileName) {
    ImageIcon icon = new ImageIcon(fileName);
    Image img = icon.getImage();
    Image newimg = img.getScaledInstance(150, 150,  java.awt.Image.SCALE_SMOOTH);
    icon = new ImageIcon(newimg);
    super.setIcon(icon);
}
```

The new code will scale your image to 150x150 pixels. Add more photos to your current directory, modify the imageFiles[ ] of ImageButton class and add the file names to it. Finally run the program.

Part 3:

You can expand your knowledge by modifying and polishing this simple application. However, it is enough for the purpose of this tutorial, and the rest is left for your own curiosity.
Zip and upload all source files into your studentId.zip file and upload to tuL7.

Mark: 6