

Screenshot-Based Guide Maker with Java

Although the author and publisher have made every effort to ensure that the information in this writing was correct at press time, the author and publisher do not assume and hereby disclaim any liability to any party for any loss, damage, or disruption caused by errors or omissions, whether such errors or omissions result from negligence, accident, or any other cause.

advertisement

The Lone Video-Maker Guide

Make professional quality online videos - all on your own.

[Click here](#)

A screen capture software allows us to easily create tutorials. Using Java, we can also create a simple app for making screenshot-based guide. It can be used, for instance, to create an Excel tutorial that explains the process of adding all the numbers in a range of cells. The output is in the form of html page instead of video.

Here is the complete code:

```
import java.awt.FlowLayout;
import java.awt.Rectangle;
import java.awt.Robot;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowStateListener;
import java.awt.image.BufferedImage;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Iterator;
import javax.imageio.ImageIO;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.SwingUtilities;

public class ScreenshotGuideMaker{
    private JFrame frame;
    private JButton bCapture;
    private JButton bCreate;
    private ArrayList<String> fName;
    private boolean stopCapture;

    public ScreenshotGuideMaker(){
        frame = new JFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(new FlowLayout());
        frame.setSize(400,200);
        frame.setResizable(false);

        bCapture = new JButton('Capture');
        bCapture.addActionListener(new ButtonHandler());
    }
}
```

```

frame.add(bCapture);

bCreate = new JButton('Create');
bCreate.addActionListener(new ButtonHandler());
frame.add(bCreate);

frame.pack();
frame.setVisible(true);
frame.addWindowStateListener(new FrameHandler());

fName = new ArrayList();           //put all images in the array
stopCapture = false;
}

class ButtonHandler implements ActionListener{
    @Override
    public void actionPerformed(ActionEvent e){
        if(e.getSource() == bCapture){
            stopCapture = false;
            MyRunnable myRunnable = new MyRunnable();
            Thread tIMG = new Thread(myRunnable);
            tIMG.start();
        }else if(e.getSource() == bCreate){
            //clear duplicate images
            if(!fName.isEmpty()){
                try {
                    clearImages();
                    createPage();
                } catch (IOException ex) {
                }
            }
        }
    }
}

private void clearImages() throws IOException{
    //start comparison and deletion
    BufferedImage bufimage = null;
    BufferedImage bufimage2 = null;

    ArrayList<String> fClone = new ArrayList();
    Iterator<String> iterator = fName.iterator();
    while(iterator.hasNext()){
        fClone.add((String)iterator.next());
    }

    for(int i = 0; i < fName.size()-1; i++){
        try {
            System.out.println(fName.get(i)+' '+fName.get(i+1));
            bufimage = ImageIO.read(new File(fName.get(i)));
            bufimage2 = ImageIO.read(new File(fName.get(i+1)));

            int width = bufimage.getWidth();
            int height = bufimage.getHeight();

            boolean same = true;
            for(int y = 0;y < height;y++){
                for(int x = 0;x < width;x++){
                    if(bufimage.getRGB(x,y) != bufimage2.getRGB(x,y)){
                        System.out.println(bufimage.getRGB(x,y) +' '+ bufimage2.getRGB(x,y));
                        same = false;
                        break;
                    }
                }
            }
            if(same == false){
                break;
            }
        }

        if(same == true){

```

```

        int idx = fClone.indexOf(fName.get(i));
        fClone.remove(idx);        //remove duplicate
    }
} catch (Exception x) {

}

}

fName.clear();
fName = fClone;
}

private void createPage() throws IOException{
    BufferedWriter bw = null;
    try{
        bw = new BufferedWriter(new FileWriter('tutorial.html'));
        bw.write('<html>');
        bw.newLine();
        bw.write('<body>');
        bw.newLine();
        bw.write('<div align=\'center\'><h2>Excel Sum Function</h2></div>');
        bw.newLine();
        for(int i = 0;i < fName.size(); i++){
            bw.write('<p align=\'center\'>');
            bw.write('<img src=' + fName.get(i) + ' width=\'50\' height=\'50\'>');
            bw.write('</p>');
            bw.newLine();
        }
        bw.newLine();
        bw.write('</body>');
        bw.newLine();
        bw.write('</html>');
    }catch(IOException e){

}finally{
    if(bw != null){
        bw.close();
    }
}
}

class MyRunnable implements Runnable {
    public void run() {
        try{
            frame.setState(JFrame.ICONIFIED);
            Robot robot = new Robot();
            Rectangle rect = new Rectangle(Toolkit.getDefaultToolkit().getScreenSize());
            BufferedImage bufImage = null;

            int i=0;
            while(stopCapture == false){
                bufImage = robot.createScreenCapture(rect);
                File file = new File('images/img'+i+'.jpg');
                ImageIO.write(bufImage, 'jpg', file);
                fName.add(file.getPath());
                i++;
                Thread.sleep(500);        //capture every 0.5 sec
            }
            frame.setState(JFrame.NORMAL);
        } catch (Exception ex) {

        }
    }
}

class FrameHandler implements WindowStateListener{
    @Override
    public void windowStateChanged(WindowEvent e) {
        if(frame.getState()==JFrame.NORMAL){
            stopCapture=true;
        }
    }
}

```

```

    }
}

public static void main(String[] args){
    SwingUtilities.invokeLater(new Runnable(){
        @Override
        public void run() {
            ScreenshotGuideMaker ap;
            ap = new ScreenshotGuideMaker();
        }
    });
}
}

```

Explanation

The interface contains two buttons. One is used to start capturing the screen (*bCapture*), and the other is used to create the web page (*bCreate*).

```

private JButton bCapture;
private JButton bCreate;

```

The array *fName* stores the image file names. It is used to remove duplicate images. Only unique images will be displayed on the web page.

```

private ArrayList<String> fName;

```

The capturing process will start or stop based on the value of the variable *stopCapture*. If it is assigned 'true', the process continues. The variable is assigned 'false' initially.

```

private boolean stopCapture;

```

The following block of code is executed when the button (*bCapture*) is clicked. An instance of 'Thread' is created. The capturing process starts.

```

if(e.getSource() == bCapture){
    stopCapture = false;
    MyRunnable myRunnable = new MyRunnable();
    Thread tIMG = new Thread(myRunnable);
    tIMG.start();
}

```

In the *run* method, you can see that the frame is minimized when the process begins.

```

frame.setState(JFrame.ICONIFIED);

```

The frame will be restored when the process ends.

```

frame.setState(JFrame.NORMAL);

```

The process is done through the use of 'Robot' class. The entire screen is captured and the screenshots are saved in a directory. The variable *bufImage* temporarily stores the image before being saved to the directory.

```

Robot robot = new Robot();
Rectangle rect = new Rectangle(Toolkit.getDefaultToolkit().getScreenSize());
BufferedImage bufImage = null;

```

As mentioned before, the process continues as long as the variable *stopCapture* is 'false'. The screen is captured every 0.5 sec.

```

int i=0;
while(stopCapture == false){
    bufImage = robot.createScreenCapture(rect);
    File file = new File('images/img'+i+'.jpg');
    ImageIO.write(bufImage, 'jpg', file);
    fName.add(file.getPath());
}

```

```

        i++;
        Thread.sleep(500);           //capture every 0.5 sec
    }

```

The function of the inner class *FrameHandler* is to assign 'true' to the variable *stopCapture* when the window/frame is restored. This will stop the capturing process.

```

class FrameHandler implements WindowStateListener{
    @Override
    public void windowStateChanged(WindowEvent e) {
        if(frame.getState()==JFrame.NORMAL){
            stopCapture=true;
        }
    }
}

```

In the following code, two functions, *clearImages()* and *createPage()*, are called when the button *bCreate* is clicked. As the name implies, the function *clearImages()* clears duplicated images while the function *createPage()* creates a web page based on the images captured.

```

else if(e.getSource() == bCreate){
    //clear duplicate images
    if(!fName.isEmpty()){
        try {
            clearImages();
            createPage();
        } catch (IOException ex) {

        }
    }
}

```

In the function *clearImages()*, images are compared pixel by pixel. Two 'BufferedImage' objects are used for this purpose.

```

BufferedImage bufimage = null;
BufferedImage bufimage2 = null;

```

The function of the iteration below is to copy the content of the array *fName* to a temporary array *fClone*. When duplicate images are found, the corresponding item in the array *fClone* is removed.

```

ArrayList<String> fClone = new ArrayList();
Iterator<String> iterator = fName.iterator();
while(iterator.hasNext()){
    fClone.add((String)iterator.next());
}

```

The process to remove a duplicated image is done through this code:

```

int idx = fClone.indexOf(fName.get(i));
fClone.remove(idx);           //remove duplicate

```

The content of the temporary array *fClone* is then brought back to the array *fName*.

```

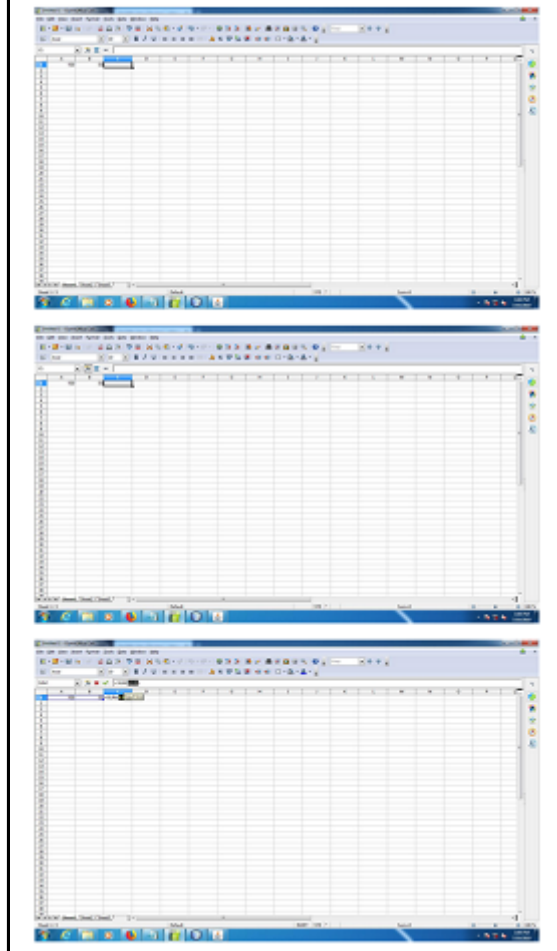
fName.clear();
fName = fClone;

```

The function *createPage()* is self explanatory. Using 'BufferedWriter', a html page is created based on the screenshots.

Here is the output of the program:

Excel Sum Function



advertisement

The Lone Video-Maker Guide

Make professional quality online videos - all on your own.

[Click here](#)

www.liberpaper.com