Starting and Stopping Animated GIF

Source: http://coding.derkeiler.com/Archive/Java/comp.lang.java.gui/2003−11/0484.html

---

From: Rhino (*rhino1_at_NOSPAM.sympatico.ca*)
Date: 11/20/03

Date: Thu, 20 Nov 2003 13:29:02 −0500

I posted this on comp.lang.java.programmer last week but nobody answered. I probably should have posted to comp.lang.java.gui in the first place; can anyone give me a hint on how to solve this problem?

___
I need some help with something. It seems like it should be easy but I've spent several hours researching via Google Usenet searches and trying things without full success.
I am trying to write an application (NOT an applet) that displays an animated GIF and also uses the mouse to stop/start the animation. The animated GIF is a single file that contains several "frames" rather than a series of discrete files, each with a single still in it.
I borrowed most of the code from a post by Linda Radecke so I know I'm "stealing" from the best ;−) My code is posted below; please forgive the ugly formatting which Outlook Express inflicts on us both. As it stands now, the program displays the animated GIF just fine although it doesn't have the code necessary to stop and start the animation yet; that's where I'm running into trouble.
I think I'm on the right track with the MouseAdapter logic but everything I've tried with respect to my stopAnimation() and stopAnimation() methods has failed so I've yanked that code out again or commented it. I've tried using Threads and (Swing) Timers in various ways but neither worked for me. Can anyone sketch in the code I need, describe it verbally, or point me to a good example so that I can resolve this? I can't believe it's really very hard, I'm just making enough small mistakes in each approach I try that I can't quite get it to work. Please note that the user may do the mouse clicks that start/stop the animation many times during the life of the program; it won't be a case of a simple start, stop, and exit. I'm really not sure if I should be starting and stopping each time or if I should do:
o start (once)
o pause and resume (as many times as user wants)
o stop/exit (once)
Also, while I have you, I've noticed that my imageUpdate() method never gets invoked under any circumstances. Although I've read the API on ImageObserver, I'm still not really clear on how and why imageUpdate() should be invoked. From posts that I've seen, I wonder if it is possible that imageUpdate() never gets invoked because the sole file loaded by MediaTracker (ham3b.gif) is loaded so quickly that imageUpdate() is not needed? Or is there a problem in my code that keeps it from being invoked? In other words, is the failure to invoke imageUpdate() a problem that should concern me or is everything working exactly as it should?
```
package timer;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
```

```
import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.MediaTracker;
import java.awt.Toolkit;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.image.ImageObserver;
import java.net.URL;
import javax.swing.JFrame;
import javax.swing.JPanel;
public class AnimatedImagePanel extends JPanel implements ImageObserver {
    static final boolean DEBUG = true;
    Image animation;
    boolean controlWithMouse = true;
    boolean frozen = false;
    int stillWidth = 0;
    int stillHeight = 0;
    Thread animator = null;
    MediaTracker tracker = null;
public static void main(String[] args) {
    JFrame myFrame = new JFrame("Animated Image Panel");
    myFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    AnimatedImagePanel aip = new AnimatedImagePanel(Color.black, "timer/",
"ham3b.gif");
    myFrame.getContentPane().add(aip, BorderLayout.CENTER);
    myFrame.pack();
    myFrame.setVisible(true);
}
public AnimatedImagePanel(Color bgColor, String animationImageFilePath,
String animationImageFile) {
    setBackground(bgColor);
/* Get the URL for the animated image file. */
    URL animationGIF =
this.getClass().getClassLoader().getResource(animationImageFilePath+animatio
nImageFile);
/* Get the actual image file. */
    animation = Toolkit.getDefaultToolkit().getImage(animationGIF);
    try {
        tracker = new MediaTracker(this);
        tracker.addImage(animation, 0);
        tracker.waitForAll();
        }
    catch (Exception excp) {
        excp.printStackTrace();
        }
/* Determine the height and width of the panel that will be needed to
display the animated GIF. */
    stillWidth = animation.getWidth(this);
    stillHeight = animation.getHeight(this);
    if (DEBUG) System.out.println("GIF dimensions in pixels (W X H) = " +
stillWidth + " X " + stillHeight);
/* If the user wishes to control the animation with the mouse, activate this
listener. */
    if (controlWithMouse) {
        addMouseListener(new MouseAdapter() {
            public void mousePressed(MouseEvent evt) {
                if (frozen) {
                    frozen = false;
                // startAnimation();
                    }
                else {
                    frozen = true;
```

```
                // stopAnimation();
                    }
                }
            });
        } //end if
}
public void paintComponent(Graphics g) {
    if (DEBUG) System.out.println("paintComponent()");
    super.paintComponent(g);
    Graphics2D g2 = (Graphics2D) g; //Could be omitted with no harm done
(change next line from 'g2' to 'g')
    g2.drawImage(animation, 0, 0, this);
}
/*
overrides imageUpdate to control the animated gif's animation
but this program works as well without this method because of
using Mediatracker ... if you work with MediaTracker normally
it's sufficient
*/
// never gets invoked
public boolean imageUpdate(Image animatedImage, int infoflags, int x, int y)
{
    if (DEBUG) System.out.println("imageUpdate()");
    if (isShowing() && (infoflags & ALLBITS) != 0) {
        repaint();
        }
    if (isShowing() && (infoflags & FRAMEBITS) != 0) {
        repaint();
        }
    return isShowing();
}
public Dimension getPreferredSize() {
    return new Dimension(stillWidth, stillHeight);
}
}
--
Rhino
---
rhino1 AT sympatico DOT ca
"If you want the best seat in the house, you'll have to move the cat."
```