


```

addWindowListener(new WindowEventHandler());
show();
animation = new Thread(this);
animation.start();
}
void setup() {
setupMenuBar();
setFont(new Font("default",Font.BOLD,18));
Toolkit toolkit = getToolkit();
frames = new Image[4];
frames[0] = toolkit.getImage("stickman1.gif");
frames[1] = toolkit.getImage("stickman2.gif");
frames[2] = toolkit.getImage("stickman3.gif");
frames[3] = toolkit.getImage("stickman4.gif");
numFrames = frames.length;
tracker = new MediaTracker(this);
for(int i=0;i<numFrames;++i)
tracker.addImage(frames[i],i);
}
void setupMenuBar() {
MenuBar menuBar = new MenuBar();
Menu fileMenu = new Menu("File");
MenuItem fileExit = new MenuItem("Exit");
fileExit.addActionListener(new MenuItemHandler());
fileMenu.add(fileExit);
menuBar.add(fileMenu);
setMenuBar(menuBar);
}
public void paint(Graphics g) {
if(allLoaded())
g.drawImage(frames[currentFrame],125,80,this);
else{
String stars = "";
for(int i=0;i<currentFrame;++i) stars += "*";
g.drawString(stars,60,60);
}
}
boolean allLoaded() {
for(int i=0;i<numFrames;++i) {
if(tracker.statusID(i,true) !=
MediaTracker.COMPLETE) return false;
}
return true;
}
public void run() {
do {
long time = System.currentTimeMillis();
if(time - lastDisplay > frameDelay) {
if(allLoaded()) {
if(fullDisplay) repaint (115,160,160,90);
else{
fullDisplay = true;
repaint();
}
}
}
}
}

```

```

}else repaint();
try {
Thread.sleep(frameDelay);
}catch(InterruptedException ex){
}
++currentFrame;
currentFrame %= numFrames;
lastDisplay = time;
}
} while (true);
}
class MenuItemHandler implements ActionListener,
ItemListener {
public void actionPerformed(ActionEvent ev){
String s=ev.getActionCommand();
if(s=="Exit"){
System.exit(0);
}
}
public void itemStateChanged(ItemEvent e){
}
}
class WindowEventHandler extends WindowAdapter {
public void windowClosing(WindowEvent e){
System.exit(0);
}
}
}
}

```

```

import java.awt.*;
import java.awt.event.*;

public class GraphicAnimationApp extends Frame
implements Runnable {
Thread animation;
int frameDelay = 100;
Image frames[];
int numFrames;
int currentFrame = 0;
long lastDisplay = 0;
int screenWidth = 400;
int screenHeight = 400;
public static void main(String args[]) {
GraphicAnimationApp app = new
GraphicAnimationApp();
}
public GraphicAnimationApp() {
super("Graphic Animation");
setup();
setSize(screenWidth,screenHeight);
addWindowListener(new WindowEventHandler());
}
}

```

```

show();
animation = new Thread(this);
animation.start();
}
void setup() {
  setupMenuBar();
  setFont(new Font("default",Font.BOLD,18));
  Toolkit toolkit = getToolkit();
  frames = new Image[4];
  frames[0] = toolkit.getImage("stickman1.gif");
  frames[1] = toolkit.getImage("stickman2.gif");
  frames[2] = toolkit.getImage("stickman3.gif");
  frames[3] = toolkit.getImage("stickman4.gif");
  numFrames = frames.length;
}
void setupMenuBar() {
  MenuBar menuBar = new MenuBar();
  Menu fileMenu = new Menu("File");
  MenuItem fileExit = new MenuItem("Exit");
  fileExit.addActionListener(new MenuItemHandler());
  fileMenu.add(fileExit);
  menuBar.add(fileMenu);
  setMenuBar(menuBar);
}
public void paint(Graphics g) {
  g.drawImage(frames[currentFrame],125,80,this);
}
public void run() {
  do {
    long time = System.currentTimeMillis();
    if(time - lastDisplay > frameDelay) {

```

```

    repaint();
    try {
      Thread.sleep(frameDelay);
    }catch(InterruptedException ex){
    }
    ++currentFrame;
    currentFrame %= numFrames;
    lastDisplay = time;
  }
} while (true);
}
class MenuItemHandler implements ActionListener,
ItemListener {
  public void actionPerformed(ActionEvent ev){
    String s=ev.getActionCommand();
    if(s=="Exit"){
      System.exit(0);
    }
  }
  public void itemStateChanged(ItemEvent e){
  }
}
class WindowEventHandler extends WindowAdapter {
  public void windowClosing(WindowEvent e){
    System.exit(0);
  }
}
}
}
}

```