

用J2ME在移动设备上实现动画的实例讲解 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/252/2021\\_2022\\_\\_E7\\_94\\_A8J2ME\\_E5\\_9C\\_A8\\_c104\\_252529.htm](https://www.100test.com/kao_ti2020/252/2021_2022__E7_94_A8J2ME_E5_9C_A8_c104_252529.htm) 使用MIDP(Mobile Information Device Profile)的开发人员经常会抱怨用些什么办法才可以在一个MIDlet上显示动画。MIDP 1.0 没有直接提供对动画的支持(正在开发中的MIDP 2.0支持)，但真要是自己去实现，其实也并非是一件很难的事。任何动画的最基本的前提，是要在足够快的时间内显示和更换一张张的图片，让人的眼睛看到动的画面效果。图片必须按照顺序画出来。从一张图片到下一张图片之间的变化越小，效果会越好。首先要做的，是使用你的图片处理软件（比如ps或者firework）创建一系列相同大小的图片来组成动画。每张图片代表动画一帧。你需要制作一定数量的帧 - - 越多的帧会让你的动画看上去越平滑。制作好的图片一定要保存成PNG(Portable Network Graphics) 格式，MIDP唯一支持的图片格式；（有两个办法让你刚做好的图片在MIDlet上变成动画。第一，把图片都放到一个web服务器上，让MIDlet下载他们，MIDP内置的HTTP支持。第二个办法更简单，把图片用MIDlet打包成jar文件。如果你使用的是J2ME开发工具，把PNG文件放在你的项目文件里面就可以了。动画的过程其实更像帐本记录：显示当前帧，然后适当地更换到下一帧。那么使用一个类来完成这个工作应该是很恰当的，那好，我们就先定义一个AnimatedImage类：

```
import java.util.*; import javax.microedition.lcdui.*; // 定义了一个动画，该动画其实只是一系列相同大小的图片 // 轮流显示，然后模拟出的动画 public class AnimatedImage extends TimerTask {
```

```

private Canvas canvas. private Image[] images. private int[][]
clipList. private int current. private int x. private int y. private int w.
private int h. // Construct an animation with no canvas. public
AnimatedImage( Image[] images ){ this( null, images, null ). }. //
Construct an animation with a null clip list. public AnimatedImage(
Canvas canvas, Image[] images ){ this( canvas, images, null ). }. //
Construct an animation. The canvas can be null, // but if not null
then a repaint will be triggered // on it each time the image changes
due to a timer // event. If a clip list is specified, the image is // drawn
multiple times, each time with a different // clip rectangle, to simulate
transparent parts. public AnimatedImage( Canvas canvas, Image[]
images, int[][] clipList ){ this.canvas = canvas. this.images = images.
this.clipList = clipList. if( images != null amp. clipList != null ){ if(
clipList.length 0 ){ w = images[0].getWidth(). h =
images[0].getHeight(). }. }. // Move to the next frame, wrapping if
necessary. public void advance( boolean repaint ){ if( current >=
images.length ){ current = 0. }. if( repaint amp. canvas != null amp.
canvas.isShown() ){ canvas.repaint( x, y, w, h ).
canvas.serviceRepaints(). }. }. // Draw the current image in the
animation. If // no clip list, just a simple copy, otherwise // set the
clipping rectangle accordingly and // draw the image multiple times.
public void draw( Graphics g ){ if( w == 0 || h == 0 ) return. int
which = current. if( clipList == null || clipList[which] == null ){
g.drawImage( images[which], x, y, g.TOP | g.LEFT ). }. else { int cx
= g.getClipX(). int cy = g.getClipY(). int cw = g.getClipWidth(). int
ch = g.getClipHeight(). int[] list = clipList[which]. for( int i = 0. i 3

```

你实例化一个AnimatedImage对象的时候你必须给AnimatedImage类的构造方法传一个Image对象数组，该数组代表动画的每一帧。使用的所有图片必须具有相同的高度和宽度。用Image.createImage()方法从jar文件里面加载图片：  
private Image[] loadFrames( String name, int frames ) throws IOException {  
Image[] images = new Image[frames]. for( int i = 0. i  
你也可以传递一个Canvas对象（可选），和一个剪辑列表（clip list）。如果你指定了一个canvas和使用一个timer来自动更换到动画的下一帧，就如下面的例子代码中一样，canvas在动画向前滚动以后自动被重画(repaint)。不过这样的实现办法是可选的，你可以这样做，也可以让程序选择合适的时候重画canvas。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)