

J2ME游戏中如何使用层 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/289/2021_2022_J2ME_E6_B8_B8_E6_88_8F_c104_289964.htm 2.0中提供

了javax.microedition.lcdui.game包，这样我们可以更容易的进行游戏开发，在这个包内一共包含了5个类，分别

是GameCanvas，Layer，LayerManager，TitledLayer和Sprite。Layer是个抽象类，TiledLayer和Sprite都是Layer的子类，前者是为了绘画场景后者是为了绘画游戏的角色的。

而LayerManager则是管理层。本文通过一个简单的例子介绍如何使用层。我们可以把游戏的空间想象成三维的，我们眼睛看到的是x-y的2维空间，事实上还存在第三维---层

。LayerManager则负责管理这些层，按照添加的顺序,LayerManager维护着层的索引，第一个添加的层的索引为0,第二个添加的层索引为1,依次类推。如果我们删除或者添加层的话，那么索引会自动调整的。这和RMS中的ID的概念是不同的，ID不是索引，只是简单的标记纪录。我们想添加层的时候只需要调用方法append(Layer l)，当然通过insert(Layer l,int index)你也可以把层插入到指定索引位置。删除层只需要remove(Layer l)。在LayerManager中一个重要的概念就是View Window，View Window控制着用户可视的区域，他的位置是相对于LayerManager的坐标系统的。通过改变可视窗口的位置我们就可以制作出屏幕滚动的效果，在后面的文章会有所介绍。通过方法setViewWindow(int x,int y,int width,int height)我们可以设置可视窗口的位置和大小。例如:通过调用方法setViewWindow(52,11,85,85)我们可以把大小为85*85，相

对LayerManager的坐标为(52,11)的一片区域显示给用户。具体定义请参考Java doc。下面我们通过一个实例演示如何使用层的概念。我们要实现把以上两图按需要显示在屏幕上，通过按键可以控制精灵的运动。这时候我们使用的Sprite类，通过它我们可以轻松制作出多帧的效果。构造器Sprite(Image image, int frameWidth, int frameHeight)可以对指定图片image按照指定的framewidth和frameheight进行分割，比如上图则可以通过下面的操作得到5帧，帧是从0开始计数的，分割的顺序为从左到右、从上到下。 Image playerImage =

```
Image.createImage("/transparent.png"); playerSprite = new Sprite  
(playerImage,32,32).
```

```
System.out.println(playerSprite.getRawFrameCount()); 我们可以  
通过调用 setFrame(int index) 方法选择指定的帧，当方  
法 paint(Graphics g) 调用的时候当前帧会被画出来。下面给出
```

```
程序的源代码。 import javax.microedition.midlet.*; import  
javax.microedition.lcdui.*; public class  
ExampleLayerManagerMidlet extends MIDlet { private Display  
display; public void startApp() { try { display =  
Display.getDisplay(this); ExampleGameCanvas gameCanvas = new  
ExampleGameCanvas(); gameCanvas.start();  
display.setCurrent(gameCanvas); } catch (Exception ex) {  
System.out.println(ex); } } public Display getDisplay() { return  
display; } public void pauseApp() { } public void  
destroyApp(boolean unconditional) { exit(); } public void exit() {  
System.gc(); destroyApp(false); notifyDestroyed(); } } import  
javax.microedition.lcdui.*; import javax.microedition.lcdui.game.*;
```

```
public class ExampleGameCanvas extends GameCanvas implements
Runnable { private boolean isPlay. // Game Loop runs when isPlay is
true private long delay. // To give thread consistency private int
currentX, currentY. // To hold current position of the ' X ' private
int width. // To hold screen width private int height. // To hold
screen height 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
```