2022年6月8日 15:42:05

# java拼图游戏

[java拼图游戏](#java拼图游戏)  
 [使用方法](#使用方法)  
 [详细设计](#详细设计)  
 [用到的包：](#用到的包)  
 [JLable类](#jlable类)  
 [其他swing类](#其他swing类)  
 [鼠标事件](#鼠标事件)  
 [总体思路：](#总体思路)  
 [MyButton类：](#mybutton类)  
 [界面展示](#界面展示)  
 [初始界面：](#初始界面)  
 [简单界面：](#简单界面)  
 [一般界面](#一般界面)  
 [困难界面](#困难界面)  
 [苦难界面](#苦难界面)  
 [关键代码理解](#关键代码理解)  
 [简单模式下的代码](#简单模式下的代码)  
 [移动的关键方法说明](#移动的关键方法说明)  
 [调试和难点](#调试和难点)  
 [音乐播放](#音乐播放)  
 [目录构架](#目录构架)

## 使用方法

**备注**：

1. 可直接下载文件夹“代码”，然后执行“pintu.java”程序
2. 说明文档在README.md中

**输入**

1. 鼠标选择游戏难度，分为简单、一般、困难和变态难度；
2. 游戏过程中鼠标+点击空白格周围图片进行格子图片交换；
3. 选择开始和返回。

**输入格式**

“I-Puzzle”通过鼠标点击指定范围的图片来输入，点击范围外则无响应。

**输入举例**

用户通过点击屏幕上的图片块来使其产生移动。

输出对每项输出作出说明

**输出格式**

图片在限定区域内的移动。

**播放音乐**

AudioClip sound = Applet.newAudioClip(new File("sounds/背景音乐.wav").toURL());

但是由于只支持wav格式的音乐，.wav这种格式是无损音乐。

**输出举例**  
不同的图片块通过点击产生移动，若一个图片块周围都是与其紧密相连的图片块则点击时不会产生移动，至少有一个空位才可移动，其他图片自动补位，最终以一块完整的图片输出。

**出错处理和恢复**

1. 点击返回进行恢复；
2. 点击窗口右上角关闭窗口；
3. 打开任务管理器关闭此程序。

## 详细设计

Java拼图的设计思想：

### 用到的包：

a) Java.awt

b) Javax.swing

c) Java.awt.event

java.awt包提供了基本的java程序的GUI设计工具。主要包括下述三个概念:

* 组件--Component
* 容器--Container
* 布局管理器--LayoutManager

##### JLable类

JLabel类可以显示文本或图像。通过在显示区域中设置垂直和水平对齐来对齐标签的内容。默认情况下，标签在显示区域中垂直居中。默认情况下，纯文本标签前沿对齐; 默认情况下，仅图像标签水平居中。

##### 其他swing类

**类构造函数**

|  |  |  |
| --- | --- | --- |
| 编号 | 构造函数 | 描述 |
| 1 | JLabel() | 创建一个没有图像且标题为空字符串的JLabel实例。 |
| 2 | JLabel(Icon image) | 使用指定的图像创建JLabel实例。 |
| 3 | JLabel(Icon image, int horizontalAlignment) | 使用指定的图像和水平对齐创建JLabel实例。 |
| 4 | JLabel(String text) | 使用指定的文本创建JLabel实例。 |
| 5 | JLabel(String text, Icon icon, int horizontalAlignment) | 使用指定的文本，图像和水平对齐创建JLabel实例。 |
| 6 | JLabel(String text, int horizontalAlignment) | 使用指定的文本和水平对齐方式创建JLabel实例。 |



**java.awt有创建用户接口、绘图和图像的所有类。**用户接口对象，例如按钮或滚动条，

在AWT(Abstrat Window Toolkit)中被称为组件， Component类是所有AWT组件的根。

用户与组件交互操作时，一些组件会激发事件， AWTEvent类及其子类用于表达AWT组件能够激发的事件。

容器是一个可以含有组件和其他容器的组件，

容器还可以有一个布局管理器，用于控制组件在容器中的位置。

AWT包含有几种布局管理器类和一个可以用来创建自己的布局管理器的接口。

在java.awt包中，又含有11个子包：

1)java.awt.color

该包提供了用于颜色的类。类中一个颜色空间的实现，

该实现基于国际颜色联盟(International Color

Consortium，简称ICC)的格式规范(版本3.4)

2)Java.awt.datatransfer

该包提供了在应用程序之间或之中传送数据的接口和类。

该包定义了一个“可传递”对象的概念，“可传递”对象通过实现Transferable接口来标识自己为可传递。

另外，它还提供了一个剪切板机制，剪切板是一个临时含有一个可传递对象的对象，

通常用于复制和粘贴操作。尽管可以在应用程序中创建一个剪切板，

大多数应用程序一般都使用系统剪切板来确保数据能够在不同平台的应用程序之间传递。

3)Java.awt.dnd

拖放(drag-and-drop)出现在许多图形用户接口的系统中。

它用手势在逻辑上表示数据或对象在两个实体之间的传递。在Windows操作系统中经常使用到这种操作，非常直观明了。

java.awt.dnd包提供了一些接口和类用于支持拖放(drag-and-drop)操作，

其定义了拖的源(drag-and-drop)和放的目标(drop-target)以及传递拖放数据的事件，

并对用户执行的操作给出可视的问馈。

4)java.awt.event

该包提供处理不同种类事件的接口和类，这些事件由AWT组件激发。

事件由事件源激发，事件监听者登记事件源，并接收事件源关于特定类型事件的通知。

Java.awt.event包定义了事件、事件监听者和事件监听者适配器。使用事件监听者适配器，更加容易编写事件监听者。

5)java.awt.font

该包提供与字体(font)相关的类和接口。

6)java.awt.geom

该包提供Java

2D类，用于定义和执行与二维几何相关的对象上的操作。

7)java.awt.im

该包提供一些类和一个输入法框架接口。该框架使得所有的文本编辑组件能够接收日文、

中文和韩文的输入法的输入，输入法让用户使用键盘上有限的键输入成千上万个不同的字符，

文本编辑组件可以使用java.awt.geom包和java.awt.event中相关类支持不同语言的输入

法。同时，框架还支持其他语言的输入法或者其他输入方式，例如手写或语音识别。

8)java.awt.im.spi

该包提供一些接口，用于支持可以在任何Java运行时环境中使用的输入法的开发，

输入法是一个让用户输入文本的软件组件，通常用于输入日文、中文和韩文。同时，

还可以用于开发其他语言的输入法以及其他方式的输入，例如手写或语音识别。

9)java.awt.image

该包提供创建和修改图像的类。

10)java.awt.image.renderable

该包提供一些类和接口，用于生成与表现无关的图像。

11)java.awt.print

java.awt.print提供一些类和接口，用干普通的打印API，该API包括

指定文档类型的能力。

页面设置和页面格式控制的机制。

管理任务控制对话框的能力。

##### 鼠标事件

### 总体思路：

### MyButton类：

该游戏主要通过按钮图片的转换进行，但是JButton本身不具有我们所需要的功能，所以创建一个MyButton类，通过继承JButton来增加我们的功能。具体功能如下：

* MyButton():构造函数，设置原始坐标xy；
* Getxy():返回原始坐标
* Pic():对按钮对象进行贴图
* Setnxy():设置现在的坐标，该坐标 随按钮位置的变化而变化
* Getnxy():返回nxy
* Reicon():返回该按钮上的贴图

## 界面展示

游戏拥有五个界面：

a. 初始界面

b. 简单模式

c. 一般模式

d. 困难模式

e. 变态模式

### 初始界面：

初始界面是该游戏一开始进入的界面，通过初始界面可以选择游戏的难度即进入其他四个界面。初始界面也是main函数中唯一的语句---建立初始见面的对象。初始界面类继承自JFrame，拥有ActionListener接口。

在初始界面中通过null布局灵活设置界面中组建的布局。用四个按钮，分别对应简单、一般、困难和变态。通过actionPerformed方法，对该界面下一系列的动作进行监听并执行。

通过点击四个按钮，创建对应的四个难度类的具体对象，并且销毁初始界面，弹出游戏界面。

对于难度设置，我们从图片分割后的数量上进行设置，从3*3，4*4，55以及最终的难度11，来增加将图片完整复原后的步骤和时间。



image-20220608155250157

### 简单界面：

简单界面和其他游戏界面差不多，最上面是开始按钮和返回。中间拼图游戏的主体，下面是拼图的原图。

按下开始按钮通过timer组件进行计时。

按下返回按钮，则销毁当前界面，返回初始界面。

在拼图主体，通过随机数打乱原图的顺序。通过xy记录每个图片应该在的位置，当两个按钮交换时，实际上改变的是图片和nxy。

在游戏的判断阶段，通过对比xy是否和nxy相等来判断游戏是否完成。

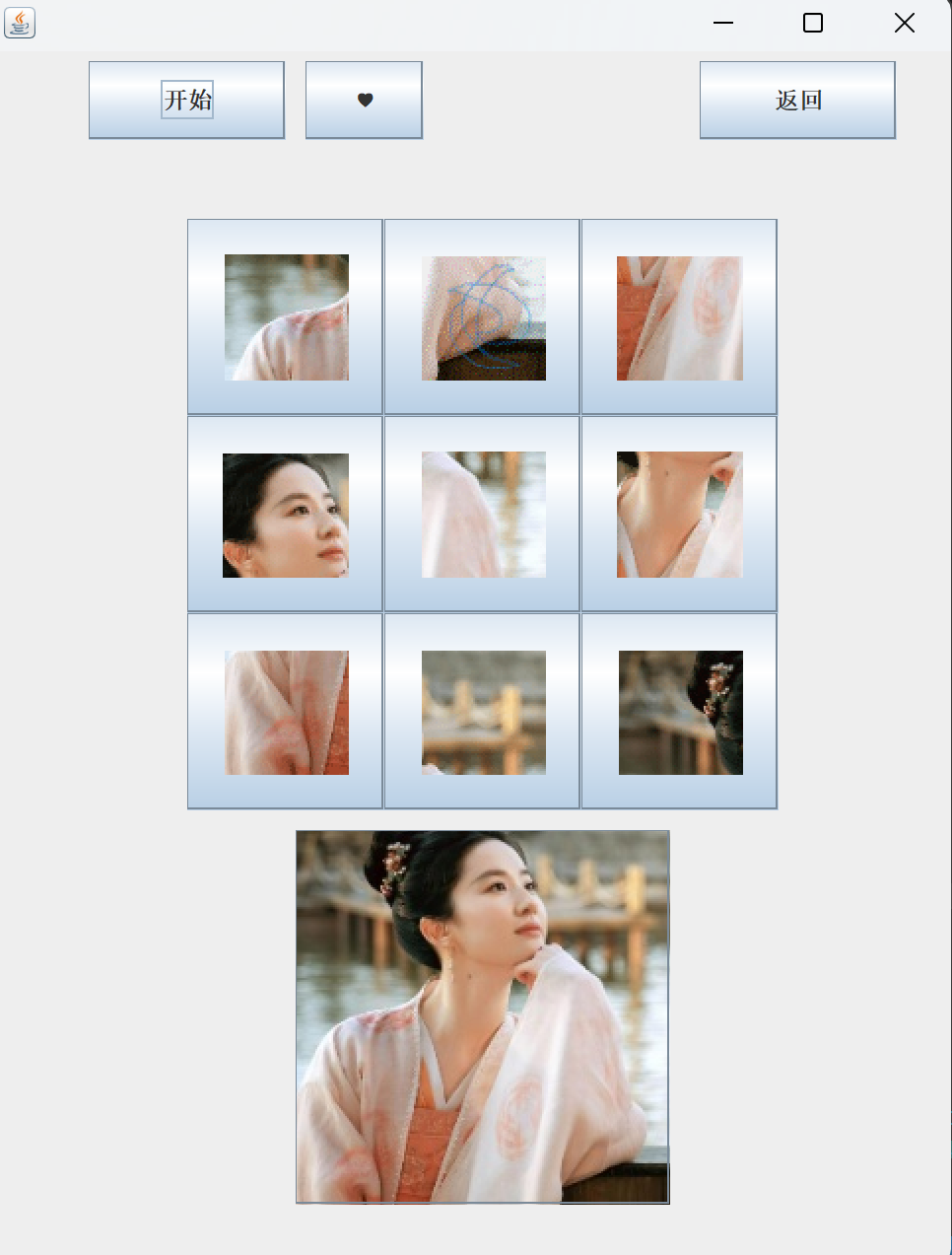


image-20220609202740291

### 一般界面

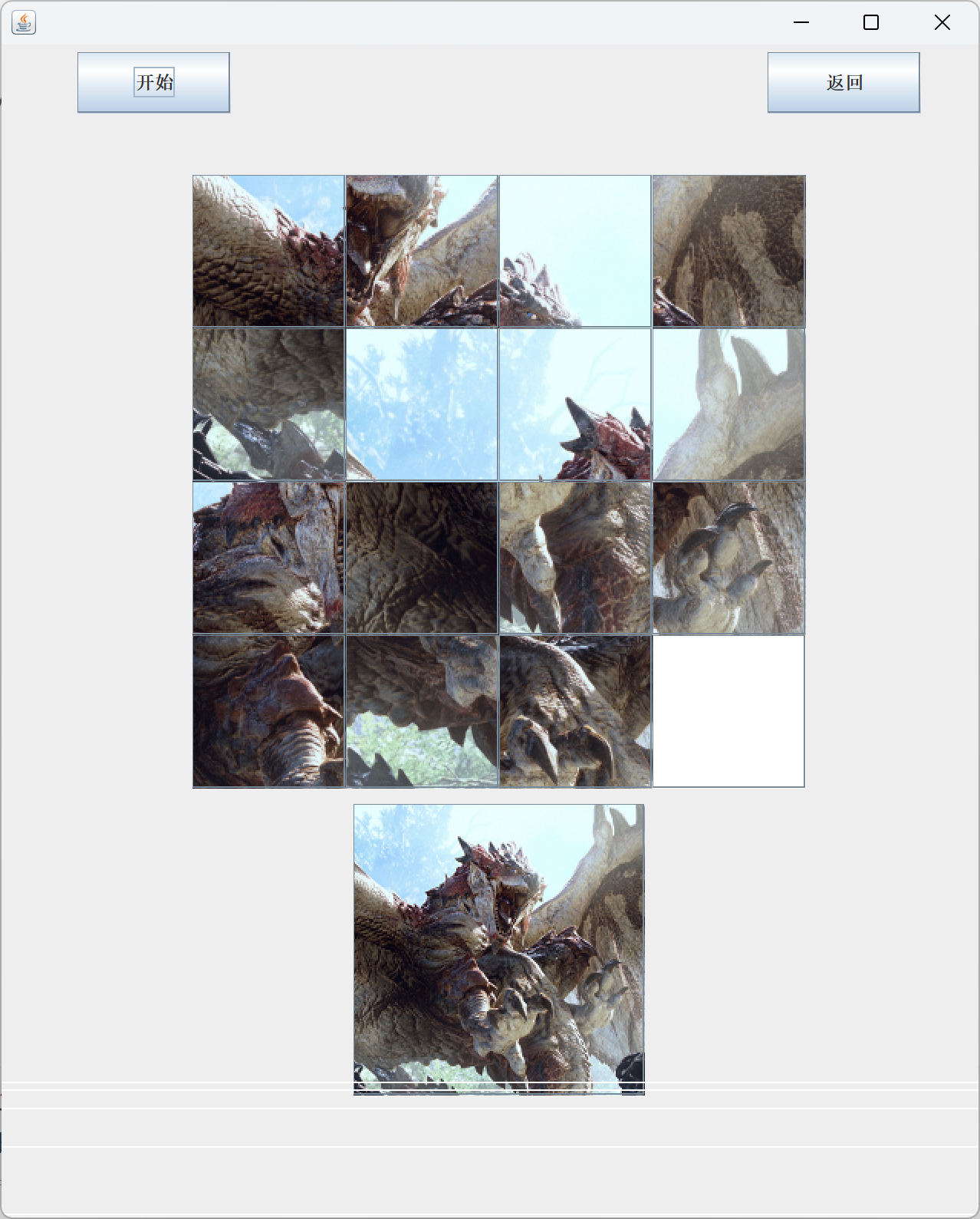


image-20220608155422292

### 困难界面

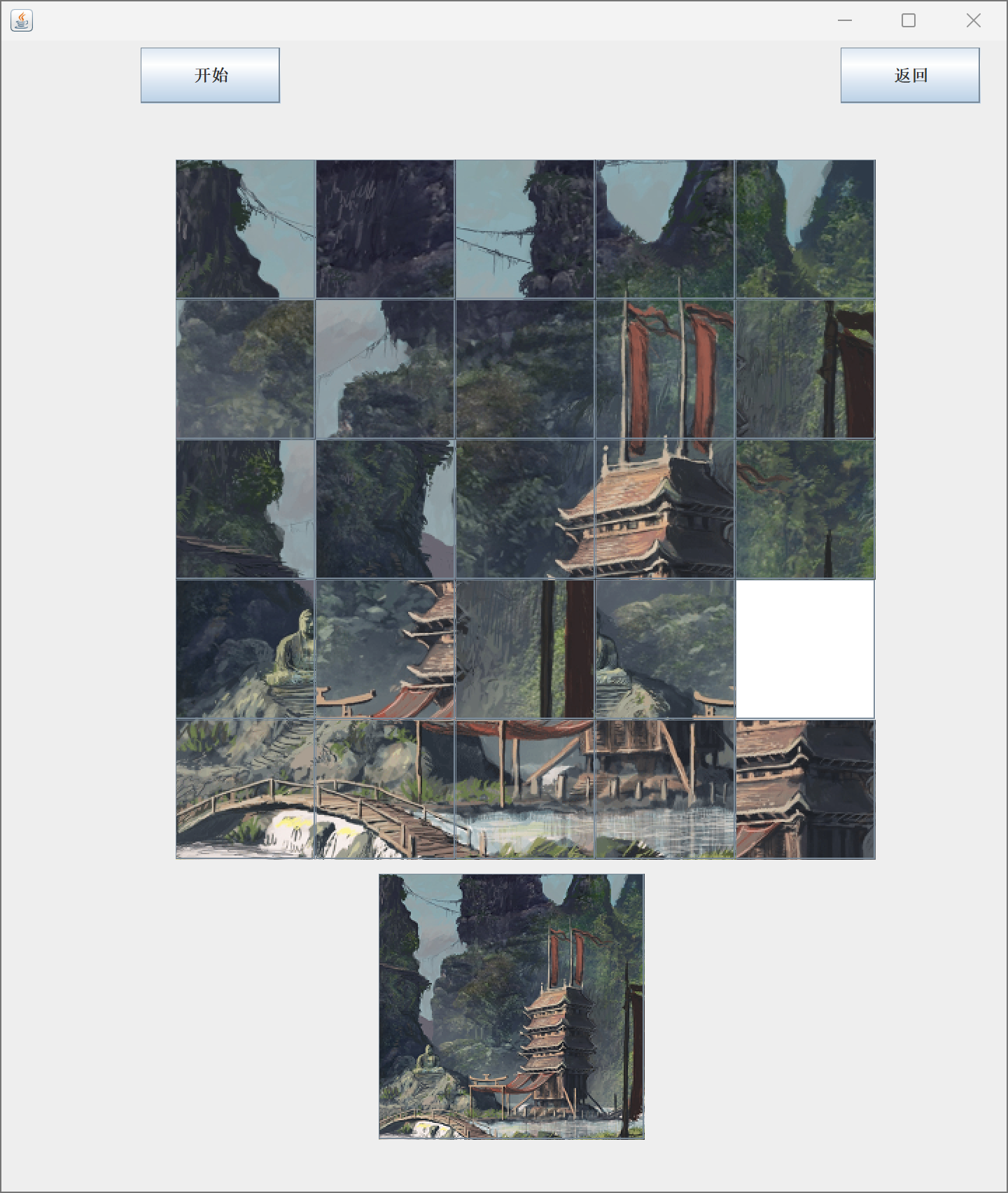


image-20220608155453301

### 苦难界面

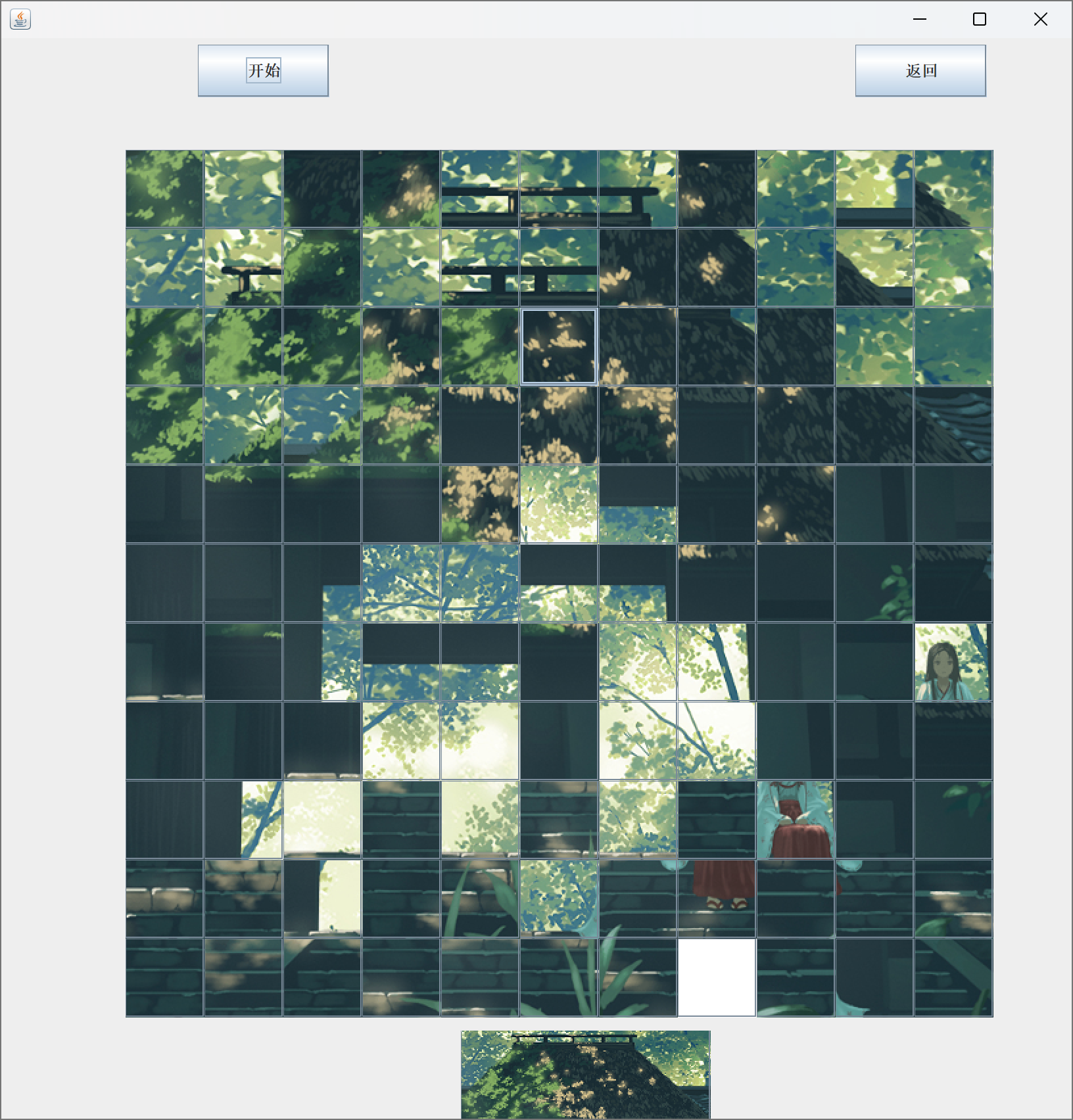


image-20220608155748409

## 关键代码理解

class FWindow extends JFrame implements ActionListener{   
 JButton jbsimple,jbmiddle,jbhigh,jbabnormal;   
 JButton music;   
 //定义简单，中级、困难、变态四个按钮选择难度   
 FWindow(){   
 setBounds(500,200,400,600);   
 setLayout(null);   
 setVisible(true);   
   
 music = new JButton("音乐");   
 //JButton定义按钮   
   
   
 jbsimple.setBounds(150,100,100,50);   
 //setBounds设置按钮的大小和位置   
   
 jbsimple.addActionListener(this);   
 //就是给前面的实例（按钮等）添加事件监听接口，   
 //this表示当前类的对象，在一个类里，你不需要new他的实例就直接可以用this调用它的方法和属性   
   
 add(jbsimple);   
 validate();   
 setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);   
 }

**在FWindow类中继承来自父类JFrame，并且实现一个接口ActionListener**

**setDefaultCloseOperation(JDialog.DISPOSE\_ON\_CLOSE);的使用方法：**

设置用户在此窗体上发起 "close" 时默认执行的操作。必须指定以下选项之一：

* DO\_NOTHING\_ON\_CLOSE（在 WindowConstants 中定义）：不执行任何操作；要求程序在已注册的WindowListener 对象的 windowClosing 方法中处理该操作。
* HIDE\_ON\_CLOSE（在 WindowConstants 中定义）：调用任意已注册的 WindowListener 对象后自动隐藏该窗体。
* DISPOSE\_ON\_CLOSE（在 WindowConstants 中定义）：调用任意已注册 WindowListener 的对象后自动隐藏并释放该窗体。
* EXIT\_ON\_CLOSE（在 JFrame 中定义）：使用 System exit 方法退出应用程序。仅在应用程序中使用。

默认情况下，该值被设置为 HIDE\_ON\_CLOSE。

**validate(); 确保组件具有有效的布局。此类主要适用于在 Container 实例上进行操作。**

[**可以参考这个文章**](https://baike.baidu.com/item/Validate/7694335)

**dispose()的使用：**

释放由此 Window、其子组件及其拥有的所有子组件所使用的所有本机屏幕资源。即这些 Component 的资源将被破坏，它们使用的所有内存都将返回到操作系统，并将它们标记为不可显示。 通过随后对 pack 或 show 的调用重新构造本机资源，可以再次显示 Window 及其子组件。重新创建的 Window 及其子组件的状态在移除 Window 的点上与这些对象的状态将是一样的（不考虑这些操作之间的其他更改）。

### 简单模式下的代码

简单界面和其他游戏界面差不多，最上面是开始按钮和返回。中间拼图游戏的主体，下面是拼图的原图。

按下开始按钮通过timer组件进行计时。

按下返回按钮，则销毁当前界面，返回初始界面。

在拼图主体，通过随机数打乱原图的顺序。通过xy记录每个图片应该在的位置，当两个按钮交换时，实际上改变的是图片和nxy。

在游戏的判断阶段，通过对比xy是否和nxy相等来判断游戏是否完成。

**Java的swing布局设置：**

当某一个类继承JFrame类是可以使用：

this.setLayout(null);

然后可以通过setBounds（）；  
对组件进行自定义大小和位置设置。  
setBounds()有四个参数：  
第一个参数改组件在JFrame中的x坐边  
第二个参数改组件在JFrame中的y坐标  
第三个参数改组件在JFrame中的组件宽度  
第四个参数改组件在JFrame中的组件高度

class SimWindow extends JFrame implements ActionListener{   
 //ActionListener创建一个类继承JFrame实现监听接口，并且重写监听接口的唯一方法   
 MyJButton b[]; //定义数组存放   
 JButton bstart,returns; //定义开始和返回   
 Timer time; //定义时间组件进行记录时间   
 JLabel lab1;   
 JButton vip; //定义vip渠道   
 int numt=1;   
 SimWindow(){ //无参构造   
 setBounds(500,200,500,650);   
 //分别对应的是Bounds的x,y,宽度和高度   
 setVisible(true);   
   
 time = new Timer(1000,this);   
 //设定时间限制以秒为基本单位，可以设置为毫秒   
   
 JPanel jpa = new JPanel();   
 add(jpa);   
 jpa.setBounds(0,0,500,45);   
 jpa.setLayout(null);   
 lab1 = new JLabel();   
 lab1.setHorizontalAlignment(JLabel.CENTER);   
 lab1.setFont(new Font("宋体",Font.BOLD,20));   
 bstart = new JButton("开始");   
 vip = new JButton("强行拆")   
 returns = new JButton("返回");   
 bstart.setBounds(50,5,100,40); //开始   
 vip.setBounds(200,5,100,40); //拆   
 returns.setBounds(360,5,100,40); //vip   
 lab1.setBounds(210,5,100,40); //   
 jpa.add(bstart);   
 jpa.add(lab1);   
 jpa.add(returns);   
   
   
 JPanel jpb = new JPanel();   
 add(jpb);   
 jpb.setBounds(100,45,300,350);   
 jpb.setLayout(null);   
 b = new MyJButton[9];   
 int l=0,t=40;   
 for(int i=1;i<=9;i++){   
 b[i-1] = new MyJButton(i-1);   
 jpb.add(b[i-1]);   
 b[i-1].setnxy(i-1);   
 b[i-1].pic("sim\\"+String.valueOf(""+i)+".gif");   
 b[i-1].setBounds(l,t,100,100);   
 b[i-1].addActionListener(this);   
 l+=100;   
   
 if(i%3==0){   
 t+=100;   
 l=0;   
 }   
 }   
 begin();   
 JPanel jpc = new JPanel();   
 add(jpc);   
 jpc.setBounds(0,350,500,200);   
 jpc.setLayout(null);   
 MyJButton yuantu = new MyJButton(33);   
 yuantu.pic("sim\\22.gif");   
 jpc.add(yuantu);   
 yuantu.setBounds(155,395,190,190);   
   
 bstart.addActionListener(this);   
 returns.addActionListener(this);   
   
   
 validate();   
 setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);   
 }   
   
 int Upp(int i){   
 if(i-3<0) return i;   
 else{   
 int temp;   
 temp = b[i-3].getnxy();   
 b[i-3].setnxy(b[i].getnxy());   
 b[i].setnxy(temp);   
 String itemp;   
 itemp = b[i-3].reicon();   
 b[i-3].pic(b[i].reicon());   
 b[i].pic(itemp);   
 return i-3;   
 }   
 }   
   
 int Doo(int i){   
 if(i+3>8) return i;   
 else{   
 int temp;   
 temp = b[i+3].getnxy();   
 b[i+3].setnxy(b[i].getnxy());   
 b[i].setnxy(temp);   
 String itemp;   
 itemp = b[i+3].reicon();   
 b[i+3].pic(b[i].reicon());   
 b[i].pic(itemp);   
 return i+3;   
 }   
 }   
   
 int Lee(int i){   
 if((i-1)/3 != i/3) return i;   
 else{   
 int temp;   
 temp = b[i-1].getnxy();   
 b[i-1].setnxy(b[i].getnxy());   
 b[i].setnxy(temp);   
 String itemp;   
 itemp = b[i-1].reicon();   
 b[i-1].pic(b[i].reicon());   
 b[i].pic(itemp);   
 return i-1;   
 }   
 }   
   
 int Rii(int i){   
 if((i+1)/3 != i/3) return i;   
 else{   
 int temp;   
 temp = b[i+1].getnxy();   
 b[i+1].setnxy(b[i].getnxy());   
 b[i].setnxy(temp);   
 String itemp;   
 itemp = b[i+1].reicon();   
 b[i+1].pic(b[i].reicon());   
 b[i].pic(itemp);   
 return i+1;   
 }   
 }   
   
 void begin(){   
 for(int i=66;i>=0;i--){   
 int ri = i%4;   
 int rz = (int)(Math.random()\*9);   
 switch(ri){   
 case 0:   
 rz=Upp(rz);   
 case 1:   
 rz=Doo(rz);   
 case 2:   
 if(rz-1>=0)   
 rz=Lee(rz);   
 case 3:   
 if(rz+1<=8)   
 rz=Rii(rz);   
 }   
 }   
 }   
   
 boolean isOK(){   
 for(int i=0;i<=8;i++){   
 if(b[i].getnxy()!=b[i].getxy())   
 return false;   
 }   
 return true;   
 }   
   
 void End(boolean bl){   
 if (bl==true){   
 time.stop();   
 lab1.setText("成功了!");   
 }   
 }   
   
 public void actionPerformed(ActionEvent e){   
 if(e.getSource()==time){   
 lab1.setText(""+numt);   
 numt++;   
 }else if(e.getSource()==bstart){   
 time.start();   
 }else if(e.getSource()==returns){   
 new FWindow();   
 dispose();   
 }else{   
 MyJButton JTemp = (MyJButton)e.getSource();   
 if(JTemp.getxy()-1>=0){   
 if(b[JTemp.getxy()-1].getnxy()==8){   
 Rii(JTemp.getxy()-1);   
 End(isOK());   
 return ;   
 }   
 }   
 if(JTemp.getxy()-3>=0){   
 if(b[JTemp.getxy()-3].getnxy() == 8){   
 Doo(JTemp.getxy()-3);   
 End(isOK());   
 return ;   
 }   
 }   
 if(JTemp.getxy()+3<=8){   
 if(b[JTemp.getxy()+3].getnxy() == 8){   
 Upp(JTemp.getxy()+3);   
 End(isOK());   
 return ;   
 }   
 }   
 if(JTemp.getxy()+1<=8){   
 if(b[JTemp.getxy()+1].getnxy() == 8){   
 Lee(JTemp.getxy()+1);   
 End(isOK());   
 return ;   
 }   
 }   
 }   
 }   
}

### 移动的关键方法说明

* MyButton():构造函数，设置原始坐标xy；
* Getxy():返回原始坐标
* Pic():对按钮对象进行贴图
* Setnxy():设置现在的坐标，该坐标 随按钮位置的变化而变化
* Getnxy():返回nxy
* Reicon():返回该按钮上的贴图

##### upp：往上移动

* + 此时i必须要>3(如果在第一行不能往上走)
  + 移动手段(通过临时的变量保存移动后的位置),最后返回的是当前的位置-3
    - 或者Go方式：x,y = y,x：
  + int temp;  
    temp = b[i-3].getnxy();  
    b[i-3].setnxy(b[i].getnxy());  
    //将原始的坐标设置为现在上移后的坐标  
    b[i].setnxy(temp);  
    String itemp;  
    itemp = b[i-3].reicon();  
    //返回按钮的贴图  
    b[i-3].pic(b[i].reicon());  
    //对按钮的对象进行贴图（移动位置）  
    b[i].pic(itemp);  
    return i-3;

##### Doo：往下移动（和上面一样）

* + 移动的代码
  + int temp;  
    temp = b[i+3].getnxy();  
    b[i+3].setnxy(b[i].getnxy());  
    b[i].setnxy(temp);  
    String itemp;  
    itemp = b[i+3].reicon();  
    b[i+3].pic(b[i].reicon());  
    b[i].pic(itemp);  
    return i+3;  
    }

##### Lee ：((i-1)/3 != i/3)左移

* + 移动的代码
  + int temp;  
    temp = b[i-1].getnxy();  
    b[i-1].setnxy(b[i].getnxy());  
    b[i].setnxy(temp);  
    String itemp;  
    itemp = b[i-1].reicon();  
    b[i-1].pic(b[i].reicon());  
    b[i].pic(itemp);  
    return i-1;

##### Rii：右移动

* + 移动的代码
  + int temp;  
    temp = b[i+1].getnxy();  
    b[i+1].setnxy(b[i].getnxy());  
    b[i].setnxy(temp);  
    String itemp;  
    itemp = b[i+1].reicon();  
    b[i+1].pic(b[i].reicon());  
    b[i].pic(itemp);  
    return i+1;

##### begin:

* + 关键代码
  + for(int i=66;i>=0;i--){  
     int ri = i%4; //2  
     int rz = (int)(Math.random()\*9);  
     //Java中Math类的random()方法可以生成[0,1)之间的随机浮点数。\*9表示产生到9的数  
     switch(ri){  
     case 0:  
     rz=Upp(rz);  
     case 1:  
     rz=Doo(rz);  
     case 2:  
     if(rz-1>=0)  
     rz=Lee(rz);  
     case 3:  
     if(rz+1<=8)  
     rz=Rii(rz);  
     }  
    }
  + 使用ipython测试ri
  + In [14]: for x in range(66,0,-1):   
     ...: print(x%4,end=",")   
     ...:   
    2,1,0,3,2,1,0,3,2,1,0,3,2,1,0,3,2,1,0,3,2,1,0,3,2,1,0,3,2,1,0,3,2,1,0,3,2,1,0,3,2,1,0,3,2,1,0,3,2,1,0,3,2,1,0,3,2,1,0,3,2,1,0,3,2,1,

##### isOK：判断

* **返回的原始坐标和返回的x,y相同，那么返回true**
* if(b[i].getnxy()!=b[i].getxy())  
   return false;

public void actionPerformed(ActionEvent e){   
 //事件响应器   
 // 组件发生事件的时候，会将事件包装成一个actionevent对象，也就是这里的e   
 // 从e里可以获得事件源对象。   
 if(e.getSource()==time){   
 lab1.setText(""+numt);   
 numt=numt+1;   
 //记时装备   
 if(numt >= 100){   
 lab1.setText("超时了!");   
 numt = 999;   
 }   
 }else if(e.getSource()==bstart){ //开始   
 time.start();   
 }else if(e.getSource()==returns){ //结束   
 new FWindow();   
 dispose();   
 }else{   
 MyJButton JTemp = (MyJButton)e.getSource();   
 if(JTemp.getxy()-1>=0){   
 //设置向左移动   
 if(b[JTemp.getxy()-1].getnxy()==8){   
 Rii(JTemp.getxy()-1);   
 End(isOK());   
 return ;   
 }   
 }

**每一次移动后都调用End进行判断，如果isok返回true那么结束**

## 调试和难点

**图片的尺寸设置和修改，我们选择最近很火的梦华录电视剧刘亦菲的图片**

[**在线剪辑网站https://www.gaitubao.com/**](https://www.gaitubao.com/)



FrRYqUmguC2414BosBiuJAhhwURy

**此时再使用lightroom的切片工具将图片剪辑为九分**

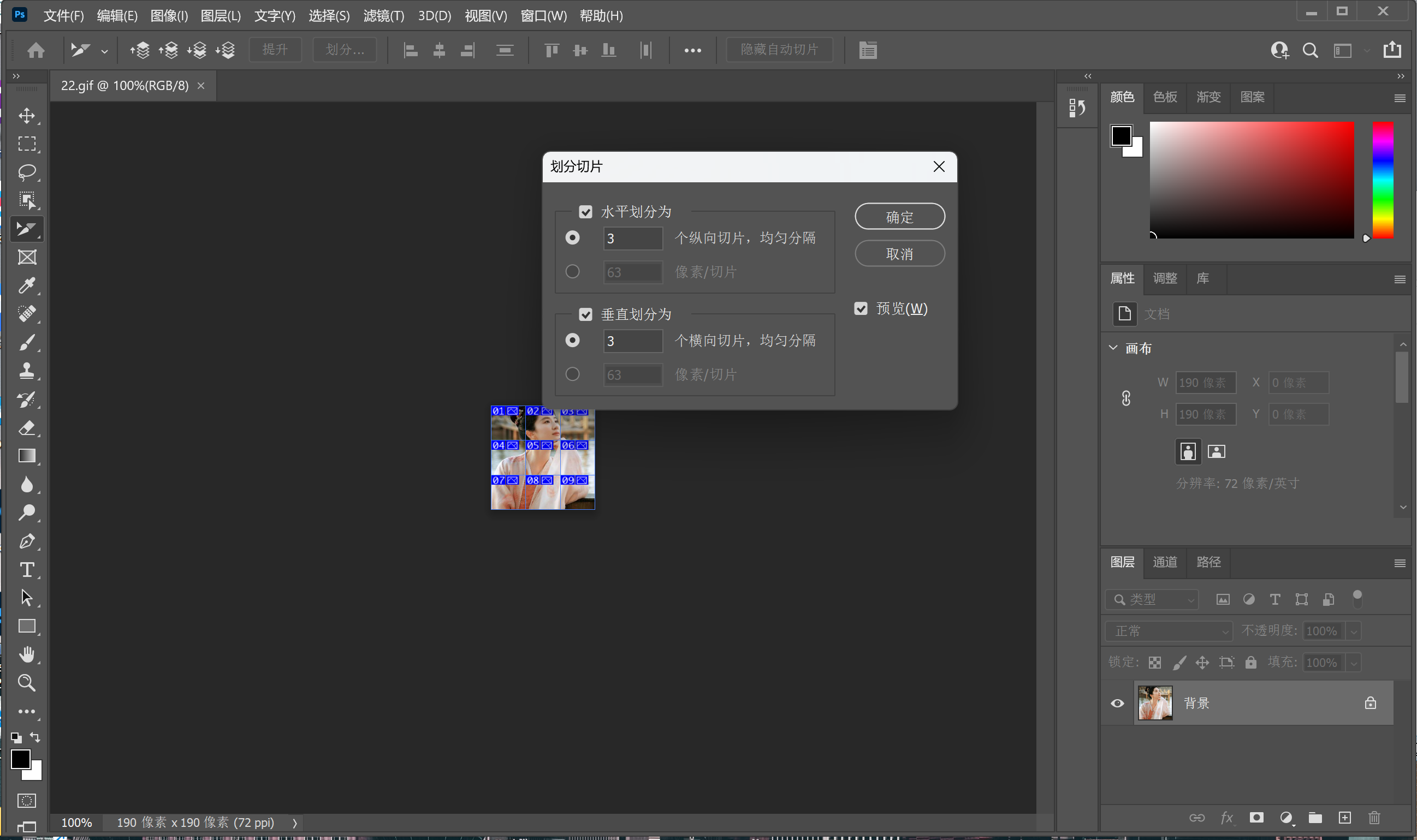


image-20220609163928556

**导入为web模式。**

**市面上找不到一个合适的对Java游戏窗体进行音乐播放**

import java.applet.Applet;   
import java.applet.AudioClip;   
import java.io.File;   
import java.net.URI;   
import java.net.URL;   
import java.nio.charset.MalformedInputException;   
   
public class MusicFrame {   
 public static void main(String[] args){   
 try{   
 File f = new File("houlaidewomen.wav");   
 URI uri = f.toURI();   
 URL url = uri.toURL();   
 AudioClip audioClip = Applet.newAudioClip(url);   
 audioClip.play();   
 }catch (MalformedInputException e){   
 e.printStackTrace();   
 }   
   
 }   
}

**在编译过程中，发现方块不能移动，想了很久**

if(b[JTemp.getxy()-1].getnxy()==8){   
 Rii(JTemp.getxy()-1);   
 //左右移动   
 End(isOK());   
 return ;   
}

**这个是用来移动代码块的**

* Rii(JTemp.getxy()-1); 左移动
* Doo(JTemp.getxy()-3); 下移动
* Upp(JTemp.getxy()+3); 上移动
* Lee(JTemp.getxy()+1); 右移动

**调用散片拼图的方法**

其实可以用一张图片使用代码切割为多个小图片，但是感觉好麻烦，就使用的是已经切割好的图片对其进行随机拼凑

b[i-1].pic("sim\\"+String.valueOf(""+i)+".gif");

##### 音乐播放

音乐播放一直是个问题，所以我选择了一个很特殊的解决方案哈哈，具体看代码

## 目录构架

**readme.md是说明文档**

* music存放音乐
* mid、abn、sim、hig存放各个模式图片
* sounds存放音乐

卷 系统 的文件夹 PATH 列表  
卷序列号为 DE95-1D97

C:.   
└─代码（执行）   
 ├─abn   
 ├─hig   
 ├─mid   
 ├─music   
 └─sim

目录: C:\Users\smile\Desktop\javae\课设\-pintu\代码（执行）

Mode LastWriteTime Length Name

d----- 2022/5/26 15:53 abn   
d----- 2022/5/26 15:53 hig   
d----- 2022/5/26 15:53 mid   
d----- 2022/6/8 15:39 music   
d----- 2022/5/26 15:53 sim   
-a---- 2022/5/26 15:53 4407 AbnWindow.class   
-a---- 2022/5/26 15:53 1528 FWindow.class   
-a---- 2022/5/26 15:53 1417258 game.jar   
-a---- 2022/5/26 15:53 4380 HigWindow.class   
-a---- 2022/5/26 15:53 90 MANIFEST.MF   
-a---- 2022/5/26 15:53 4380 MidWindow.class   
-a---- 2022/5/26 15:53 788 MyJButton.class   
-a---- 2022/5/26 15:53 285 pintu.class   
-a---- 2022/5/26 22:43 18348 pintu.java   
-a---- 2022/5/26 15:53 4380 SimWindow.class