**Java程序设计**

**课程设计报告**

**\*\*\*\***

|  |  |
| --- | --- |
| **专 业** |  |
| **班 级** |  |
| **学 号** | **\*\*\*\*\*\*\*\*\*\*** |
| **姓 名** |  |

**二 ○ 年 月**

**目录**

1. ···················································设计目的
2. ·············································设计思路与分析

·················································2.1时序图

·················································2.2类图

1. ···················································具体代码
2. ·····················································评分表

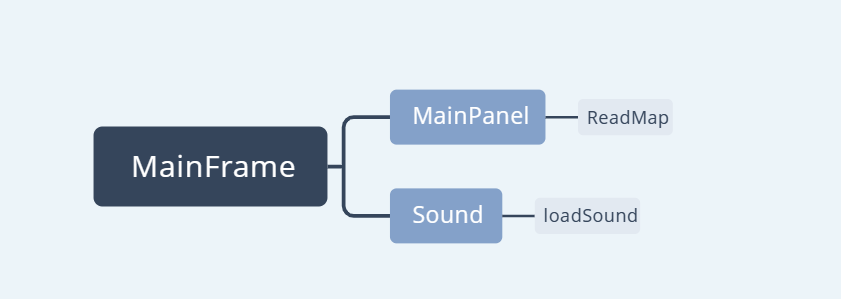
## 设计目的

设计一个，简单好玩的推箱子游戏。

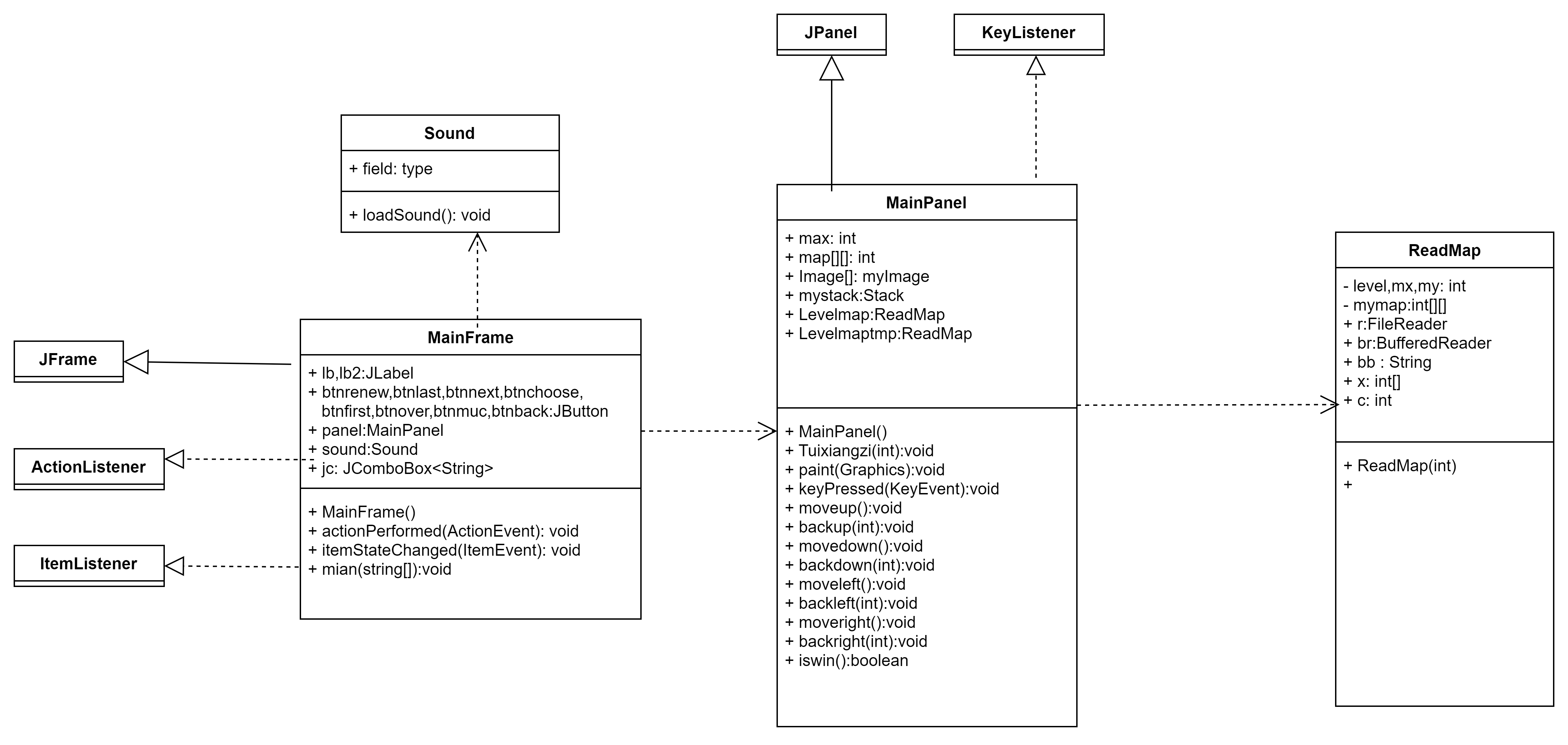
## 二、设计思路与分析

1. 查找网上旧版推箱子的关卡设计资料，了解设计模型
2. 编写代码，运用了java的GUI
3. 查找图片和一些音乐，让程序更加美观

## 2.1时序图



## 2.2 类图



## 三、系统设计与实现

## 3.1部分功能截图

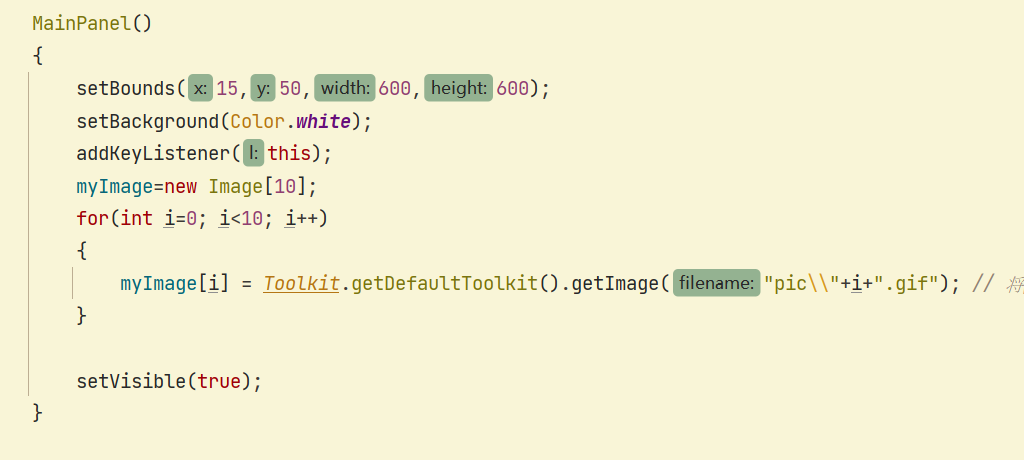
1、设置右侧按钮



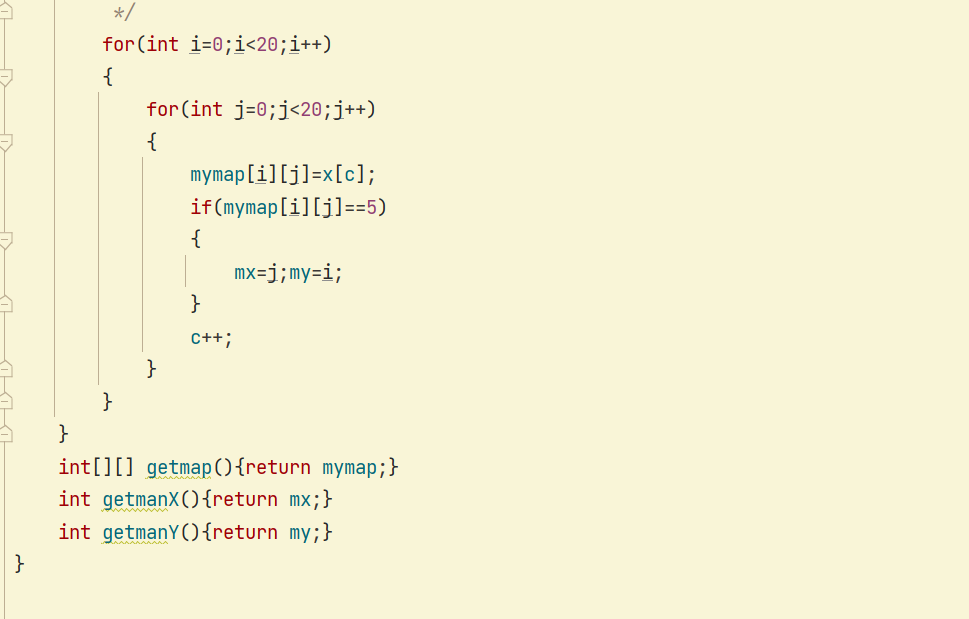
2、读取音乐序列



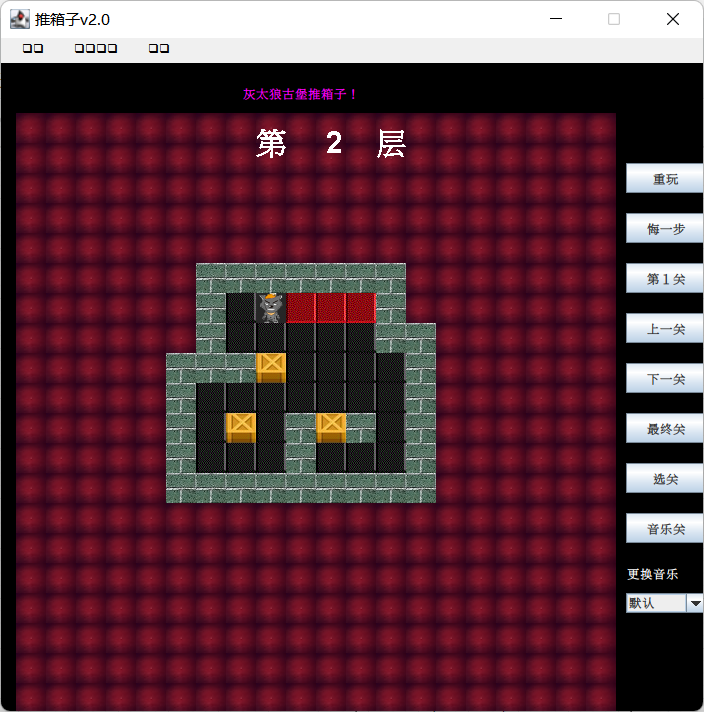
3、读取背景图



4、读取地图



5、运行结果截图



## 3.2核心代码

MainFrame.java

package TuiXiangZi;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

public class MainFrame extends JFrame implements ActionListener, ItemListener {

JLabel lb;

JLabel lb2;

JButton btnrenew,btnlast,btnnext,btnchoose,btnfirst,btnover,btnmuc,btnback;

MainPanel panel;

Sound sound;

JComboBox<String> jc=new JComboBox<>(); // 创建下拉列表框对象

MenuItem renew=new MenuItem(" 重新开始");

MenuItem back=new MenuItem(" 悔一步");

MenuItem last=new MenuItem(" 上一关");

MenuItem next=new MenuItem(" 下一关");

MenuItem choose=new MenuItem(" 选关");

MenuItem exit=new MenuItem(" 退出");

MenuItem qin=new MenuItem(" 琴萧合奏");

MenuItem po=new MenuItem(" 泡泡堂");

MenuItem guang=new MenuItem(" 灌篮高手");

MenuItem nor=new MenuItem(" 默认");

MenuItem eye=new MenuItem(" eyes on me");

MenuItem about=new MenuItem(" 关于推箱子...");

MainFrame() // 创建窗口

{

/\*

设置标题栏及按钮

\*/

super("推箱子v2.0");

setSize(720,720);

setVisible(true);

setResizable(false);

setLocation(300,20);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Container cont=getContentPane();

cont.setLayout(null);

cont.setBackground(Color.black);

Menu choice=new Menu(" 选项"); // 创建菜单

choice.add(renew);

choice.add(last);

choice.add(next);

choice.add(choose);

choice.add(back);

choice.addSeparator();

choice.add(exit);

// ActionListener用来响应用户点击按钮

renew.addActionListener(this);

last.addActionListener(this);

next.addActionListener(this);

choose.addActionListener(this);

exit.addActionListener(this);

back.addActionListener(this);

Menu setmuc=new Menu(" 设置音乐");

setmuc.add(nor);

setmuc.add(qin);

setmuc.add(po);

setmuc.add(guang);

setmuc.add(eye);

nor.addActionListener(this);

qin.addActionListener(this);

po.addActionListener(this);

guang.addActionListener(this);

eye.addActionListener(this);

Menu help=new Menu(" 帮助");

help.add(about);

about.addActionListener(this);

MenuBar bar=new MenuBar();

bar.add(choice);bar.add(setmuc);bar.add(help);

setMenuBar(bar);

/\*

设置右侧按钮

\*/

nor.setEnabled(false);

lb=new JLabel("灰太狼古堡推箱子！",SwingConstants.CENTER);

lb2=new JLabel("更换音乐",SwingConstants.CENTER);

add(lb);add(lb2);

lb.setBounds(100,20,400,20);

lb.setForeground(Color.magenta);

lb2.setBounds(625,500,55,20);

lb2.setForeground(Color.white);

btnrenew=new JButton("重玩");

btnback=new JButton("悔一步");

btnlast=new JButton("上一关");

btnnext=new JButton("下一关");

btnchoose=new JButton("选关");

btnfirst=new JButton("第１关");

btnover=new JButton("最终关");

btnmuc=new JButton("音乐关");

add(btnrenew);

add(btnlast);

add(btnnext);

add(btnchoose);

add(btnfirst);

add(btnover);

add(btnmuc);

add(btnback);

btnrenew.setBounds(625,100,80,30);

btnrenew.addActionListener(this);

btnback.setBounds(625,150,80,30);

btnback.addActionListener(this);

btnfirst.setBounds(625,200,80,30);

btnfirst.addActionListener(this);

btnlast.setBounds(625,250,80,30);

btnlast.addActionListener(this);

btnnext.setBounds(625,300,80,30);

btnnext.addActionListener(this);

btnover.setBounds(625,350,80,30);

btnover.addActionListener(this);

btnchoose.setBounds(625,400,80,30);

btnchoose.addActionListener(this);

btnmuc.setBounds(625,450,80,30);

btnmuc.addActionListener(this);

jc.setBounds(625,530,80,20);

jc.addItem("默认");

jc.addItem("琴萧合奏");

jc.addItem("泡泡堂");

jc.addItem("灌篮高手");

jc.addItem("eyes on me");

jc.addItemListener(this); // 符号监听器

cont.add(jc);

sound=new Sound();

sound.loadSound();

panel=new MainPanel();

add(panel);

panel.Tuixiangzi(panel.level);

panel.requestFocus();

validate();

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==btnrenew||e.getSource()==renew) // 重玩

{

panel.Tuixiangzi(panel.level);

panel.requestFocus();

panel.remove();

}

else if(e.getSource()==btnlast||e.getSource()==last) // 上一关

{

panel.level--;

if(panel.level<1)

{panel.level++;

JOptionPane.showMessageDialog(this,"本关是第一关");

panel.requestFocus();}

else

{

panel.Tuixiangzi(panel.level);

panel.requestFocus();

}

panel.remove();

}

else if(e.getSource()==btnnext||e.getSource()==next) // 下一关

{

panel.level++;

if(panel.level>panel.maxlevel())

{panel.level--;

JOptionPane.showMessageDialog(this,"本关已是最后一关");

panel.requestFocus();}

else

{

panel.Tuixiangzi(panel.level);

panel.requestFocus();

}

panel.remove();

}

else if(e.getSource()==exit)

System.exit(0);

else if(e.getSource()==about)

{

JOptionPane.showMessageDialog(this, "JAVA推箱子游戏！(version 3.0)");

}

else if(e.getSource()==btnchoose||e.getSource()==choose) // 选关

{

String lel=JOptionPane.showInputDialog(this,"请输入您要转到的关卡号：(1~50)");

panel.level=Integer.parseInt(lel);

panel.remove();

if(panel.level>panel.maxlevel()||panel.level<1)

{JOptionPane.showMessageDialog(this, "没有这一关！！！");panel.requestFocus();}

else

{

panel.Tuixiangzi(panel.level);

panel.requestFocus();

}

}

else if(e.getSource()==btnfirst)

{

panel.level=1;

panel.Tuixiangzi(panel.level);

panel.requestFocus();

panel.remove();

}

else if(e.getSource()==btnover)

{

panel.level=panel.maxlevel();

panel.Tuixiangzi(panel.level);

panel.requestFocus();

panel.remove();

}

else if(e.getSource()==btnmuc)

{

if(sound.isplay())

{

sound.mystop();

btnmuc.setText("音乐开");

}

else

{

sound.loadSound();

btnmuc.setText("音乐关");

}

panel.requestFocus();

}

else if(e.getSource()==btnback||e.getSource()==back) // 撤销一步

{

if(panel.isMystackEmpty())JOptionPane.showMessageDialog(this, "您还未移动！！！");

else

{

switch(panel.back())

{

case 10:panel.backup(10);break;

case 11:panel.backup(11);break;

case 20:panel.backdown(20);break;

case 21:panel.backdown(21);break;

case 30:panel.backleft(30);break;

case 31:panel.backleft(31);break;

case 40:panel.backright(40);break;

case 41:panel.backright(41);break;

}

}

panel.requestFocus();

}

/\*

监听客户端行为，并切换下拉菜单状态，符号监听器会根据状态切换音乐

\*/

else if(e.getSource()==nor)

{

jc.setSelectedIndex(0);

}

else if(e.getSource()==qin)

{

jc.setSelectedIndex(1);

}

else if(e.getSource()==guang)

{

jc.setSelectedIndex(3);

}

else if(e.getSource()==eye)

{

jc.setSelectedIndex(4);

}

else if(e.getSource()==po)

{

jc.setSelectedIndex(2);

}

}

/\*

根据客户端下拉菜单选中的按钮，切换音乐

\*/

public void itemStateChanged(ItemEvent ie)

{

int no=jc.getSelectedIndex();

switch(no)

{

case 0:sound.setMusic("nor.mid");

if(sound.isplay())

sound.mystop();

sound.loadSound();

btnmuc.setText("音乐关");

nor.setEnabled(false);

qin.setEnabled(true);

guang.setEnabled(true);

eye.setEnabled(true);

po.setEnabled(true);panel.requestFocus();break;

case 1:sound.setMusic("qin.mid");

if(sound.isplay())

sound.mystop();

sound.loadSound();

btnmuc.setText("音乐关");

nor.setEnabled(true);

qin.setEnabled(false);

guang.setEnabled(true);

eye.setEnabled(true);

po.setEnabled(true);panel.requestFocus();break;

case 2:sound.setMusic("popo.mid");

if(sound.isplay())

sound.mystop();

sound.loadSound();

btnmuc.setText("音乐关");

nor.setEnabled(true);

qin.setEnabled(true);

guang.setEnabled(true);

eye.setEnabled(true);

po.setEnabled(false);panel.requestFocus();break;

case 3:sound.setMusic("guang.mid");

if(sound.isplay())

sound.mystop();

sound.loadSound();

btnmuc.setText("音乐关");

nor.setEnabled(true);

qin.setEnabled(true);

guang.setEnabled(false);

eye.setEnabled(true);

po.setEnabled(true);panel.requestFocus();break;

case 4:sound.setMusic("eyes on me.mid");

if(sound.isplay())

sound.mystop();

sound.loadSound();

btnmuc.setText("音乐关");

nor.setEnabled(true);

qin.setEnabled(true);

guang.setEnabled(true);

eye.setEnabled(false);

po.setEnabled(true);panel.requestFocus();break;

}

}

public static void main(String[] args) {

new MainFrame();

}

}

MainPanel.java

package TuiXiangZi;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import java.util.Stack;

public class MainPanel extends JPanel implements KeyListener

{

int max=50; // 关卡数，一共50关

int[][] map; // map数组不同位置的数值不同，其值对应pic文件夹中图片的序号，并通过379行drawImage方法画到面板上，小人移动时

// 其值会根据相应逻辑更新，并通过repaint方法更新面板，由于更新逻辑类似，仅以moveup()方法为例做了注释

int[][] maptmp; // 该数组记录地图最初的情况，其值不会随小人移动而变化

int manX,manY; // 人小所在位置的坐标

Image[] myImage; // 用于存储pic文件夹下的照片

ReadMap Levelmap; // 关卡地图对象，会被getmap()方法解析为map数组对象

ReadMap Levelmaptmp; // 临时关卡地图对象，会被getmap()方法解析为maptmp数组对象

int len=30;

public int level=1; // 当前关卡数

Stack mystack=new Stack(); // 用于存储小人移动的情况，根据上下左右移动的情况，以及移动时相应路线上箱子等的情况，存在不同的

// 状态符号数，将该符号数压入栈中，以便回撤时，根据状态符号数执行相应的相反操作

MainPanel()

{

setBounds(15,50,600,600);

setBackground(Color.white);

addKeyListener(this);

myImage=new Image[10];

for(int i=0; i<10; i++)

{

myImage[i] = Toolkit.getDefaultToolkit().getImage("pic\\"+i+".gif"); // 将pic文件夹下0-9的图片读入数组

}

setVisible(true);

}

void Tuixiangzi(int i)

{

Levelmap=new ReadMap(i); // 根据关数生成地图

Levelmaptmp=new ReadMap(i);

map=Levelmap.getmap(); // 将地图转化为二维数组

manX=Levelmap.getmanX();

manY=Levelmap.getmanY(); // 获得小人初始坐标

maptmp=Levelmaptmp.getmap();

repaint(); // 根据小人移动，更新地图

}

int maxlevel(){return max;}

public void paint(Graphics g) // 将图片加载到面板上

{

for(int i=0; i<20; i++)

for(int j=0; j<20; j++)

{

g.drawImage(myImage[map[j][i]],i\*len,j\*len,this);

}

g.setColor(new Color(255,255,255));

g.setFont(new Font("楷体\_2312",Font.BOLD,30));

g.drawString("第",240,40);

g.drawString(String.valueOf(level),310,40);

g.drawString("层",360,40);

}

public void keyPressed(KeyEvent e)

{

if(e.getKeyCode()==KeyEvent.VK\_UP){moveup();}

if(e.getKeyCode()==KeyEvent.VK\_DOWN){movedown();}

if(e.getKeyCode()==KeyEvent.VK\_LEFT){moveleft();}

if(e.getKeyCode()==KeyEvent.VK\_RIGHT){moveright();}

if(iswin())

{

if(level==max){JOptionPane.showMessageDialog(this, "恭喜您通过最后一关！！！");}

else

{

String msg="恭喜您通过第"+level+"关!!!\n是否要进入下一关？";

int type=JOptionPane.YES\_NO\_OPTION;

String title="恭喜过关！";

int choice=0;

choice=JOptionPane.showConfirmDialog(null,msg,title,type);

if(choice==1)System.exit(0);

else if(choice==0)

{

level++;

Tuixiangzi(level);

}

}

mystack.removeAllElements();

}

}

public void keyTyped(KeyEvent e){}

public void keyReleased(KeyEvent e){}

boolean isMystackEmpty(){return mystack.isEmpty();}

int back(){return (Integer)mystack.pop();}

void remove(){mystack.removeAllElements();}

/\*

推箱子小人移动时，需要更新相应位置的图片，maptmp和map数组中存储的数字

会被379行 graphics.drawImage方法调用生成相应的图片，再通过repaint方法更新mainpanel

map数组值1 2 3 4 5的含义见857行

\*/

void moveup() // 向上移动

{

if(map[manY-1][manX]==2||map[manY-1][manX]==4) // 如果要移动的地方可以移动或者是箱子的目的地

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9) // 如果原始地图中该位置是箱子的目的地，或者箱子已经放在目的地了

map[manY][manX]=4; // 小人往上移动或者推箱子，这个地方会更新成箱子的目的地

else map[manY][manX]=2; // 否则表示该位置可移动

map[manY-1][manX]=8; // 将该位置更新为图片8

repaint();manY--;mystack.push(10); // 更新画板，更新Y方向位置坐标，将该操作压入栈中，且该状态符号数为10，用于回撤

}

else if(map[manY-1][manX]==3) // 如果要移动的地方是箱子

{

if(map[manY-2][manX]==4) // 如果向上两格是箱子的目的地

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

map[manY][manX]=4;

else map[manY][manX]=2; // 更新小人移动后，该位置的图片，逻辑同上

map[manY-1][manX]=8; // 更新小人图片

map[manY-2][manX]=9; // 表示箱子移动到了目的地

repaint();manY--;mystack.push(11); // 该操作状态符号数为11

}

else if(map[manY-2][manX]==2) // 如果向上两格是可以移动

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

map[manY][manX]=4;

else map[manY][manX]=2; // 更新小人移动后，该位置的图片，逻辑同上

map[manY-1][manX]=8; // 更新小人图片

map[manY-2][manX]=3; // 目前该位置是箱子

repaint();manY--;mystack.push(11);

}

else {map[manY][manX]=8;repaint();} // 否则，上两格可能是墙或者箱子，则只更新该位置的图片，由于没有有效移动，没有符号数压入栈

}

else if(map[manY-1][manX]==9) // 如果要移动的地方是箱子已在目的地

{

if(map[manY-2][manX]==4) // 如果上两格也是箱子的目的地

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

map[manY][manX]=4;

else map[manY][manX]=2; // 更新小人移动后，该位置的图片，逻辑同上

map[manY-1][manX]=8;

map[manY-2][manX]=9;

repaint();manY--;mystack.push(11);

}

else if(map[manY-2][manX]==2)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

map[manY][manX]=4;

else map[manY][manX]=2;

map[manY-1][manX]=8;

map[manY-2][manX]=3;

repaint();manY--;mystack.push(11);

}

else {map[manY][manX]=8;repaint();}

}

if(map[manY-1][manX]==1) // 如果要移动的地方是墙，则只更新该位置小人的图片

{

map[manY][manX]=8;repaint();

}

}

void backup(int t)

{

int n=t;

if(n==10) // 目的地可以移动

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

{

map[manY][manX]=4;

}

else map[manY][manX]=2;

}

else if(n==11) // 移动的是箱子

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9) // 此时的manY是移动后的坐标，对应的是moveup中manY-1

{

map[manY][manX]=9; // 因为状态符号数11表示该操作移动了箱子，所以，该位置原先箱子已在目的地

}

else map[manY][manX]=3;

if(maptmp[manY-1][manX]==4||maptmp[manY-1][manX]==9)

{

map[manY-1][manX]=4;

}

else map[manY-1][manX]=2;

}

map[manY+1][manX]=8;

repaint();manY++;

}

void movedown()

{

if(map[manY+1][manX]==2||map[manY+1][manX]==4)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

map[manY][manX]=4;

else map[manY][manX]=2;

map[manY+1][manX]=5;

repaint();manY++;mystack.push(20);

}

else if(map[manY+1][manX]==3)

{

if(map[manY+2][manX]==4)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

map[manY][manX]=4;

else map[manY][manX]=2;

map[manY+1][manX]=5;

map[manY+2][manX]=9;

repaint();manY++;mystack.push(21);

}

else if(map[manY+2][manX]==2)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

map[manY][manX]=4;

else map[manY][manX]=2;

map[manY+1][manX]=5;

map[manY+2][manX]=3;

repaint();manY++;mystack.push(21);

}

else {map[manY][manX]=5;repaint();}

}

else if(map[manY+1][manX]==9)

{

if(map[manY+2][manX]==4)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

map[manY][manX]=4;

else map[manY][manX]=2;

map[manY+1][manX]=5;

map[manY+2][manX]=9;

repaint();manY++;mystack.push(21);

}

else if(map[manY+2][manX]==2)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

map[manY][manX]=4;

else map[manY][manX]=2;

map[manY+1][manX]=5;

map[manY+2][manX]=3;

repaint();manY++;mystack.push(21);

}

else {map[manY][manX]=5;repaint();}

}

else if(map[manY+1][manX]==1)

{

map[manY][manX]=5;repaint();

}

}

void backdown(int t)

{

int n=t;

if(n==20)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

{

map[manY][manX]=4;

}

else map[manY][manX]=2;

}

else if(n==21)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

{

map[manY][manX]=9;

}

else map[manY][manX]=3;

if(maptmp[manY+1][manX]==4||maptmp[manY+1][manX]==9)

{

map[manY+1][manX]=4;

}

else map[manY+1][manX]=2;

}

map[manY-1][manX]=5;

repaint();manY--;

}

void moveleft()

{

if(map[manY][manX-1]==2||map[manY][manX-1]==4)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

map[manY][manX]=4;

else map[manY][manX]=2;

map[manY][manX-1]=6;

repaint();manX--;mystack.push(30);

}

else if(map[manY][manX-1]==3)

{

if(map[manY][manX-2]==4)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

map[manY][manX]=4;

else map[manY][manX]=2;

map[manY][manX-1]=6;

map[manY][manX-2]=9;

repaint();manX--;mystack.push(31);

}

else if(map[manY][manX-2]==2)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

map[manY][manX]=4;

else map[manY][manX]=2;

map[manY][manX-1]=6;

map[manY][manX-2]=3;

repaint();manX--;mystack.push(31);

}

else {map[manY][manX]=6;repaint();}

}

else if(map[manY][manX-1]==9)

{

if(map[manY][manX-2]==4)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

map[manY][manX]=4;

else map[manY][manX]=2;

map[manY][manX-1]=6;

map[manY][manX-2]=9;

repaint();manX--;mystack.push(31);

}

else if(map[manY][manX-2]==2)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

map[manY][manX]=4;

else map[manY][manX]=2;

map[manY][manX-1]=6;

map[manY][manX-2]=3;

repaint();manX--;mystack.push(31);

}

else {map[manY][manX]=6;repaint();}

}

else if(map[manY][manX-1]==1)

{

map[manY][manX]=6;repaint();

}

}

void backleft(int t)

{

int n=t;

if(n==30)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

{

map[manY][manX]=4;

}

else map[manY][manX]=2;

}

else if(n==31)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

{

map[manY][manX]=9;

}

else map[manY][manX]=3;

if(maptmp[manY][manX-1]==4||maptmp[manY][manX-1]==9)

{

map[manY][manX-1]=4;

}

else map[manY][manX-1]=2;

}

map[manY][manX+1]=6;

repaint();manX++;

}

void moveright()

{

if(map[manY][manX+1]==2||map[manY][manX+1]==4)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

map[manY][manX]=4;

else map[manY][manX]=2;

map[manY][manX+1]=7;

repaint();manX++;mystack.push(40);

}

else if(map[manY][manX+1]==3)

{

if(map[manY][manX+2]==4)

{

if(maptmp[manY][manX]==4)

map[manY][manX]=4;

else map[manY][manX]=2;

map[manY][manX+1]=7;

map[manY][manX+2]=9;

repaint();manX++;mystack.push(41);

}

else if(map[manY][manX+2]==2)

{

if(maptmp[manY][manX]==4)

map[manY][manX]=4;

else map[manY][manX]=2;

map[manY][manX+1]=7;

map[manY][manX+2]=3;

repaint();manX++;mystack.push(41);

}

else {map[manY][manX]=7;repaint();}

}

else if(map[manY][manX+1]==9)

{

if(map[manY][manX+2]==4)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

map[manY][manX]=4;

else map[manY][manX]=2;

map[manY][manX+1]=7;

map[manY][manX+2]=9;

repaint();manX++;mystack.push(41);

}

else if(map[manY][manX+2]==2)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

map[manY][manX]=4;

else map[manY][manX]=2;

map[manY][manX+1]=7;

map[manY][manX+2]=3;

repaint();manX++;mystack.push(41);

}

else {map[manY][manX]=7;repaint();}

}

else if(map[manY][manX+1]==1)

{

map[manY][manX]=7;repaint();

}

}

void backright(int t)

{

int n=t;

if(n==40)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

{

map[manY][manX]=4;

}

else map[manY][manX]=2;

}

else if(n==41)

{

if(maptmp[manY][manX]==4||maptmp[manY][manX]==9)

{

map[manY][manX]=9;

}

else map[manY][manX]=3;

if(maptmp[manY][manX+1]==4||maptmp[manY][manX+1]==9)

{

map[manY][manX+1]=4;

}

else map[manY][manX+1]=2;

}

map[manY][manX-1]=7;

repaint();manX--;

}

boolean iswin()

{

boolean num=false;

out: for(int i=0; i<20; i++)

for (int j = 0; j < 20; j++) {

if (maptmp[i][j] == 4 || maptmp[i][j] == 9)

if (map[i][j] == 9) num = true;

else {

num = false;

break out; // 跳出最外层循环

}

}

return num;

}

}

ReadMap.java

package TuiXiangZi;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileReader;

import java.io.IOException;

public class ReadMap {

private int level,mx,my;

private int[][] mymap=new int[20][20];

FileReader r;

BufferedReader br;

String bb="";

int[] x;int c=0;

ReadMap(int k)

{

level=k;

String s;

try

{

File f=new File("maps\\"+level+".map");

r=new FileReader(f);

br=new BufferedReader(r); // 从缓冲区中读取字符流，提高效率

}

catch (IOException e)

{

System.out.println(e);

}

try

{

while ((s=br.readLine())!=null)

{

bb=bb+s;

}

}

catch (IOException g)

{

System.out.println(g);

}

byte[] d=bb.getBytes();

int len=bb.length();

int[] x=new int[len];

for(int i=0;i<bb.length();i++)x[i]=d[i]-48; //ASCII表的零是第48个

/\*

地图中数字的含义：

0 外围修饰区域

1 墙

2 该位置箱子可移动

3 箱子

4 箱子的目的地

5 小人起始位置

\*/

for(int i=0;i<20;i++)

{

for(int j=0;j<20;j++)

{

mymap[i][j]=x[c];

if(mymap[i][j]==5)

{

mx=j;my=i;

}

c++;

}

}

}

int[][] getmap(){return mymap;}

int getmanX(){return mx;}

int getmanY(){return my;}

}

Sound.java

package TuiXiangZi;

import javax.sound.midi.MidiSystem;

import javax.sound.midi.Sequence;

import javax.sound.midi.Sequencer;

import java.io.File;

public class Sound {

String path=new String("musics\\"); // 以变量存储地址，方便日后修改

String file=new String("nor.mid");

Sequence seq;

Sequencer midi; // 负责播放MIDI序列的对象

boolean sign;

void loadSound()

{

try {

seq= MidiSystem.getSequence(new File(path+file)); // 从musics文件夹中获得MIDI序列

midi=MidiSystem.getSequencer();

midi.open();

midi.setSequence(seq);

midi.start();

midi.setLoopCount(Sequencer.LOOP\_CONTINUOUSLY); // 音乐循环播放

}

catch (Exception ex) {ex.printStackTrace();}

sign=true;

}

void mystop(){midi.stop();midi.close();sign=false;}

boolean isplay(){return sign;}

void setMusic(String e){file=e;}

}

**《java程序设计》课程设计评分表**

**学 号：**

**姓 名：**

|  |  |  |  |
| --- | --- | --- | --- |
| **项目** | **内容** | **所占分值** | **成绩** |
| **功能模块** | **功能实现** | **20分** |  |
| **可扩展性** | **10分** |  |
| **程序界面** | **10分** |  |
| **代码部分** | **面向对象** | **10分** |  |
| **算法** | **10分** |  |
| **答辩部分** | **问题** | **30分** |  |
| **报告** |  | **10分** |  |
| **总分** |  | | |