



**Java考核**

课程名称\_\_\_Java程序设计\_\_\_\_\_\_

题目名称\_\_\_趣味推箱子\_\_\_\_\_\_\_\_\_

学生学院\_\_\_计算机\_\_\_\_\_\_\_\_\_\_\_\_

专业班级\_\_ 软件工程15(2)班\_\_\_

学 号\_\_\_3115005287\_\_\_\_\_\_\_\_

学生姓名\_\_\_肖奕钦\_\_\_\_\_\_\_\_\_\_\_\_\_

指导教师\_\_\_赵锐\_\_\_ \_\_\_\_\_\_\_

2016 年12月5日

1. **课设题目**

趣味推箱子小游戏

**二、设计方案论证**

**1、设计思路**

设计一个经典的推箱子小游戏，在窗体里有墙，箱子，胜利的标志，和工人，用户可以通过键盘上的“上”、“下”、“左”、“右”按键控制工人的上下左右移动动轨迹，工人具有推箱子的功能，箱子会根据工人的运动而运动，墙起到阻碍的用，箱子碰到墙会停下来，工人碰到墙也会停下来，当所有箱子都停在胜利的标志的位置时，游戏便胜利。本趣味推箱子游戏是一个Java Application应用程序。本程序设计是基于java Swing JFream 组件 设计与实现。 创建一个主类：GrameFrame 然后通过一系列的加入板块，实现人工以及墙体的加入。最后在箱子被推到指定地点时，利用监听事件，响应出信息。下图为系统的流程图。

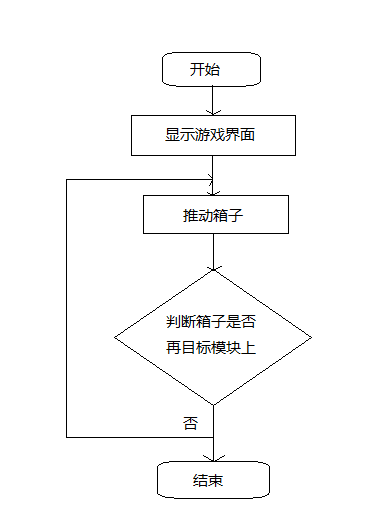


图 1 系统流程图

**2、详细设计**

开

始

**1）界面设计**

下图为程序需要用到的图片。工人上下左右移动图片，箱子图片，背景墙体图片等， 墙的设计需要用数组添加图片，其他图片添加需要SETBOUNS来设定其的位置和大小。

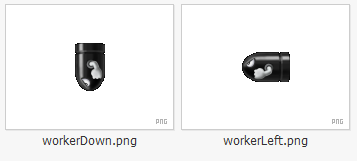
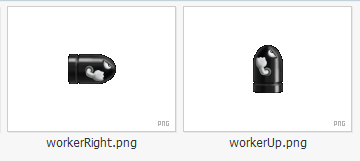


图2 程序用图

**2）系统功能模块划分**

在该系统中总共划分了三个模块

(1)初始化模块，对游戏界面进行初始化操作，对各个按钮的功能进行操作使之当动作发生时有一定的反应及相应。

(2)移动箱子模块，在推箱子游戏中最主要的模块，用户可以移动箱子到界面中的各个地方，以完成游戏设定的任务。

(3)移动小人模块，通过对工人的操作使之能上下左右移动，能够操作木块到各个位置。

**3）系统设计**

(1)Swing 中的基本控件

本项目中会分别使用JFrame、JPanel、JLabel、JOptionPane 四种控件，含义如下：

JFrame：代表窗体对象，Windows 操作系统中的一个窗体都可以看做一个JFrame 对象。JFrame 常见的操作有设置标题（setTitle）、设置窗口大小（setBounds）、设置是否显示（setVisible）等。

JPanel：即面板，面板中可以包含各种Swing 控件。JPanel 常见的操作有设置背景色（setBackground）、添加控件（add）、设置布局（setLayout）等。

JLabel：即标签，可以用来显示文字或图片。JLabel 常见的操作有设置图片（setIcon）、设置位置（setBounds）等。

JOptionPane：通常使用JOptionPane 的showMessageDialog 方法弹出对话框。

(2)GameFrame类中的方法图表

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 继承关系 | 作用 | 实现的功能 |
| GameFrame | extends JFrame | 定义所有成员，箱子，工人等等。 | 主类，包含了所有的功能。 |
| keyPressed | 没有 | 响应工人和箱子的移动以及工人、箱子、墙之间的碰撞检测。 | 响应工人和箱子的移动以及工人、箱子、墙之间的碰撞检测。 |
| keyReleased | 没有 | 完成定义 | 没有 |

表1 GameFrame类中

**4）代码实现简介**

主类:GameFrame 用于初始化各种对象.

JPanel panel = new JPanel();

JLabel worker int SPEED = 12;

int[] pos = new int[]

JLabel[] walls

JLabel boxJLabel goal

方法 GameFrame 用于完善信息.然后再进行动作事件的监听.

主要的监听对象为:

判断 工人的移动方向是否会出界

worker.setBounds(worker.getX() + xSpeed, worker.getY() + ySpeed, worker.getWidth(), worker.getHeight());

使工人可以推动箱子

在步骤2.7 的代码后继续输入如下代码使工人可以推劢箱子：

f (worker.getBounds().intersects(box.getBounds())) {

box.setBounds(box.getX() + xSpeed, box.getY() + ySpeed,

box.getWidth(), box.getHeight());

}

判断箱子是否撞到墙壁

在步骤3.2 的if 语句结束前继续输入如下代码判断箱子是否撞到墙壁：

for (int i = 0; i < walls.length; i++) {

if (box.getBounds().intersects(walls[i].getBounds())) {

worker.setBounds(worker.getX() - xSpeed, worker.getY() - ySpeed,

worker.getWidth(), worker.getHeight());

box.setBounds(box.getX() - xSpeed, box.getY() - ySpeed,

box.getWidth(), box.getHeight());

}

}

判断是否胜利

在步骤3.3 的语句后继续输入如下代码判断是否胜利：

if (box.getX()==goal.getX() && box.getY()==goal.getY()) {

JOptionPane.showMessageDialog(null, "您赢啦！");

}

**5）关键技术及算法**

本程序主要利用窗体来加载游戏，用面板来装入所有成员，墙用数组来设定它的位置，箱子、工人、墙和胜利的标志的图片均用JLabel来装载，在主类里添加键盘监听器，监听器里添加了keyPressed、keyReleased、keyTyped方法，其中keyPressed方法里实现了工人的移动，箱子的移动，箱子与工人、墙之间的碰撞检测，游戏的胜利，keyReleased、keyTyped均设置为空。

**三、设计结果与分析**

游戏运行时可以通键盘上的“上”、“下”、“左”、“右”按键移动工人，并且推动箱子。下图3为游戏运行界面。

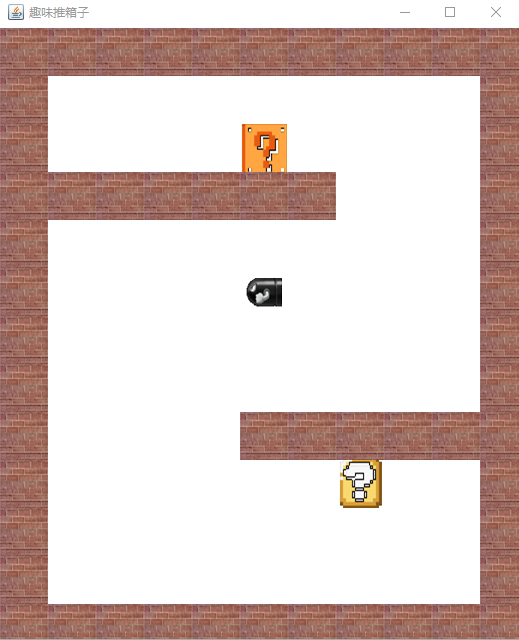


图3 游戏开始界面

当箱子与所给定的目标物重合时，游戏胜利，弹出窗口，显示“你赢了！”的字样，此时游戏结束。下图4为游戏胜利界面。

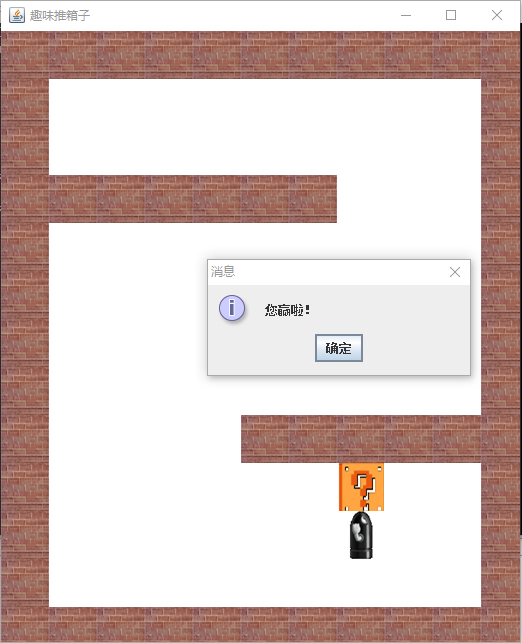


图4 游戏胜利界面

**四、设计体会**

通过编写本次课程设计程序，我完成了基于JAVA的推箱子小游戏的软件开发任务。在些其间，我遇到了许多问题，如地图文件的制作与读取、游戏界面过大或者过小、游戏图形的大小问题、工人与箱子的移动算法等等。不过，最终我都一一解决了相关问题，对其中比较主要的几点体会总结如下：

第一，为了熟悉推箱子小游戏的相关运作，我经常在电脑游戏上运用推箱子游戏，一步步地发现它所有的功能，然后再思考与查阅资料进行分析，最后经过逐步的设计与反复的修改论证，才完成了整个小游戏的开发。

第二，完成了从“小到全”，从“全到精”的转变。在设计的过程中,我经常会遇到一些困难,我也借此机会学会如何通过反复思考和测试找出自己的错误并加以改正，再进行下一个步骤的设计。本程序就是这样一点一点的完成的，在这之后再来优化整个程序。

第三，本次设计也让我得到一个不断学习新鲜事物的机会，从设计初的诸多不懂到最后能够顺利完成，期间我体会到在实践中学习的重要性，这对于将来我走上工作岗位具有重要意义。

第四，通过毕业设计使我对Java这门课程加深了解，也对我今后的就业之路奠定了一定的基础。

经过这些日子的不断努力,本次的毕业设计程序基本上完成了。回顾整个设计过程，我学到了许多书本上没有学到的知识。通过这次自己制作的软件，丰富了自己的实践技能，扩展了本专业的知识面，不仅使我受益匪浅，同时也体验到了开发软件开发的难度与不断更新自己的重要性。

我感觉学习java应该是循环渐进,有始有终,按部就班,脚踏实地.java是一门有着阶梯性的一们语言,如果要学习它.我觉得最好还是按照java的学习体系,先学习什么，在学习什么，只有这样，我们在学习中才会遇到更少的麻烦.java是一门更倾向于网络的语言.不要上来就学习android,这样不好，因为这没有基础.在以后的学习中很可能遇到很多的麻烦,这就不好了.信心,恒心,毅力是最重要的.这是我们必须具备的心理素质.要是学习这门语言开始的时候很有兴趣,遇到苦难就退缩,这样最终会放弃学习java,没有经历风雨怎么见彩虹.编程就是有的时候就是那么这么人.会遇到很多的困惑.但是一旦你弄懂了,或者是你把问题解决了，你会感到很兴奋,编程的快乐就在此了。

**附录**

import javax.swing.JFrame;

import java.awt.Color;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.ImageIcon;

import javax.swing.JLabel;

public class GameFrame extends JFrame {

JPanel panel = new JPanel();

//[2.3] 加入工人

JLabel worker = new JLabel(new ImageIcon("workerLeft.png"));

public static final int SPEED = 12;

//[2.6] 加入围墙

int[] pos = new int[]{0,0, 1,0, 2,0, 3,0, 4,0, 5,0, 6,0, 7,0, 8,0, 9,0, 10,0, 0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8, 0,9, 0,10, 0,11, 0,12, 1,12, 2,12, 3,12, 4,12, 5,12, 6,12, 7,12, 8,12, 9,12, 10,12, 10,1, 10,2, 10,3, 10,4, 10,5, 10,6, 10,7, 10,8, 10,9, 10,10, 10,11, 10,12, 1,3, 2,3, 3,3, 4,3, 5,3, 6,3, 5,8, 6,8, 7,8, 8,8, 9,8, 10,8 ,11,8 };

JLabel[] walls = new JLabel[pos.length/2];

//[3.1] 加入箱子与胜利标志

JLabel box = new JLabel(new ImageIcon("box.png"));

JLabel goal = new JLabel(new ImageIcon("goal.png"));

public GameFrame() {

//[2.2] 完善窗体

this.setTitle("趣味推箱子");

this.setBounds(200, 60, 535, 650);

this.setContentPane(panel);

panel.setLayout(null);

panel.setBackground(Color.black);

//[2.3] 加入工人

panel.add(worker);

worker.setBounds(240, 240, 48, 48);

//[2.6] 加入围墙

for (int i = 0; i < walls.length; i++) {

walls[i] = new JLabel(new ImageIcon("wall.png"));

panel.add(walls[i]);

walls[i].setBounds(pos[i \* 2] \* 48, pos[i \* 2 + 1] \* 48, 48 , 48);

}

//[3.1] 加入箱子与胜利标志

panel.add(box);

panel.add(goal);

box.setBounds(240, 96, 48, 48);

goal.setBounds(336, 432, 48, 48);

this.setVisible(true);

//[2.4] 监听键盘事件

this.addKeyListener(new KeyListener() {

//键盘按下事件

public void keyPressed(KeyEvent e) {

//[2.5] 使工人可以移动

int xSpeed = 0, ySpeed = 0;

switch (e.getKeyCode()) {

case KeyEvent.VK\_LEFT :

xSpeed = -SPEED;

worker.setIcon(new ImageIcon("workerLeft.png"));

break;

case KeyEvent.VK\_RIGHT :

xSpeed = SPEED;

worker.setIcon(new ImageIcon("workerRight.png"));

break;

case KeyEvent.VK\_UP :

ySpeed = -SPEED;

worker.setIcon(new ImageIcon("workerUp.png"));

break;

case KeyEvent.VK\_DOWN :

ySpeed = SPEED;

worker.setIcon(new ImageIcon("workerDown.png"));

break;

default:

return;

}

worker.setBounds(worker.getX() + xSpeed, worker.getY() + ySpeed, worker.getWidth(), worker.getHeight());

//[2.7] 判断工人是否撞到墙壁

for (int i = 0; i < walls.length; i++) {

if (worker.getBounds().intersects(walls[i].getBounds())) {

worker.setBounds(worker.getX() - xSpeed, worker.getY() - ySpeed, worker.getWidth(), worker.getHeight());

}

}

//[3.2] 使工人可以推动箱子

if (worker.getBounds().intersects(box.getBounds())) {

box.setBounds(box.getX() + xSpeed, box.getY() + ySpeed, box.getWidth(), box.getHeight());

//[3.3] 判断箱子是否撞到墙壁

for (int i = 0; i < walls.length; i++) {

if (box.getBounds().intersects(walls[i].getBounds())) {

worker.setBounds(worker.getX() - xSpeed, worker.getY() - ySpeed, worker.getWidth(), worker.getHeight());

box.setBounds(box.getX() - xSpeed, box.getY() - ySpeed, box.getWidth(), box.getHeight());

}

}

//[3.4] 判断是否胜利

if (box.getX()==goal.getX() && box.getY()==goal.getY()) {

JOptionPane.showMessageDialog(null, "您赢啦！");

}

}

}

public void keyReleased(KeyEvent e) {

}

public void keyTyped(KeyEvent e) {

}});

}

public static void main(String[] args) {

//[2.1]创建窗体

new GameFrame();

}

}