**课程设计(论文)**

课程名称 面向对象课程设计

学 院 信息科学与工程学院

班级学号 2203050320

学生姓名 闻家尉

指导教师 郭向坤 曹烨

2023年5月26日

**课程设计任务书及成绩**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学生姓名 | 闻家尉 | | 班级学号 | 2203050320 |
| 课程名称 | 面向对象课程设计 | | | |
| 题目 | 编写FlappyBird的Column类和BirdGame类 | | | |
| **课程设计目标与任务:**  **任务1:**  **实现管子的随机生成和移动**  **任务2:**  **创建Bird column ground生成画框并添加图片**  **实现鼠标键盘对小鸟的控制 重复执行程序直到小鸟死亡**  **工作计划与进度安排:**  **第一阶段：明确题目要求，查阅相关材料，确定设计方案。**  **第二阶段：详细设计，编写代码，调试程序。**  **第三阶段：撰写课程设计报告。**  **第四阶段：程序验收、答辩** | | | | |
| **评语：**  **成绩：** | | | | |
| 指导教师：  28798d9fb5896784a6f13f41f0ced9b  2023年6月2日 | | 专业负责人：  e01dc78a13e51fae4733e8a7bda10cc  2023年6月2日 | | 学院教学副院长：  fbe74ce8159b36ea032abb20a4d8b05  2023年6月2日 |

**目 录**

[1 飞扬小鸟游戏的界面设计 3](#_Toc135833263)

[1.1 题目内容 3](#_Toc135833264)

[1.2 程序类图 3](#_Toc135833265)

[1.3 源程序及注释 3](#_Toc135833266)

[1.4 运行结果 5](#_Toc135833267)

[2 飞扬小鸟游戏的运行设计制作 6](#_Toc135833268)

[2.1 题目内容 6](#_Toc135833269)

[2.2 类图 7](#_Toc135833270)

[2.3 源程序及注释 7](#_Toc135833271)

2.4 运行结果 ...............................................................................14

[3 总结 15](#_Toc135833272)

[参考文献 15](#_Toc135833273)

# 飞扬小鸟游戏的界面设计

## 题目内容

设置柱子长宽等属性间隔 设计随机生成函数使柱子持续生成直到游戏结束

## 1.2 程序类图

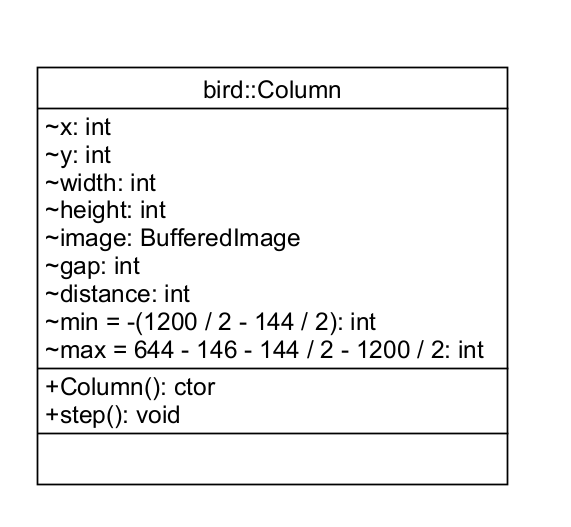


图1.1Column类的程序类图

## 1.3 源程序及注释

package bird;

import java.awt.image.BufferedImage;

//柱子类

public class Column {

int x ; // 坐标

int y ;

int width ; // 宽高

int height ;

BufferedImage image ; // 图片

int gap ; // 上下柱子之间间隙

int distance ; // 水平方向柱子之间的距离

int min = -(1200 / 2 - 144 / 2);

int max = 644 -146 - 144 / 2 - 1200 / 2;

public Column(){

image = BirdGame.column\_image;

width = image.getWidth();

height = image.getHeight();

gap = 144 ;

distance = 244;

x = 432 ;

y = (int)( Math.random() \* (max - min) + min);

}

// 柱子走一步的方法

public void step(){

x = x - 1;

if (x <= -width){ // 柱子越界

x = 432 ;

// 重新随机 纵坐标

y = (int)( Math.random() \* (max - min) + min);

}

}

}

## 1.4 运行结果



图1.2Column类的运行结果图

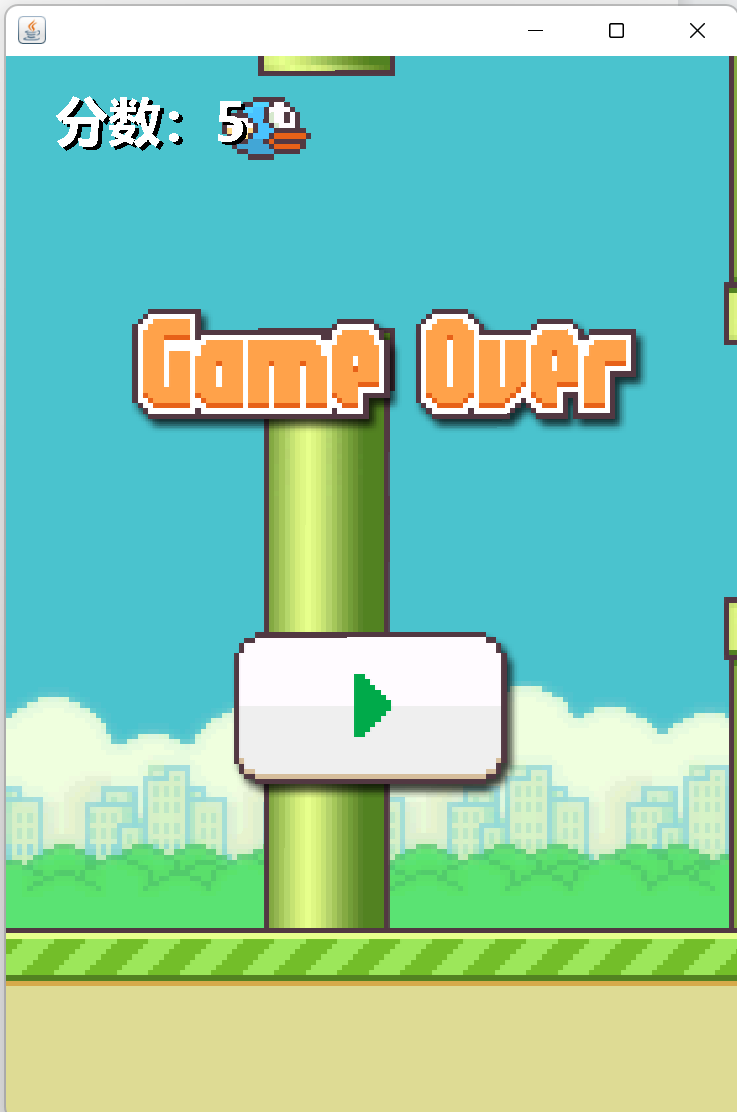


图1.3Column类的运行结果图

# 2 飞扬小鸟游戏的运行设计制作

## 2.1 题目内容

寻找图型变量并存入自定义变量中 创建画框加入图片变量

捕捉键盘鼠标的控制 循环运行类方法

## 2.2 类图

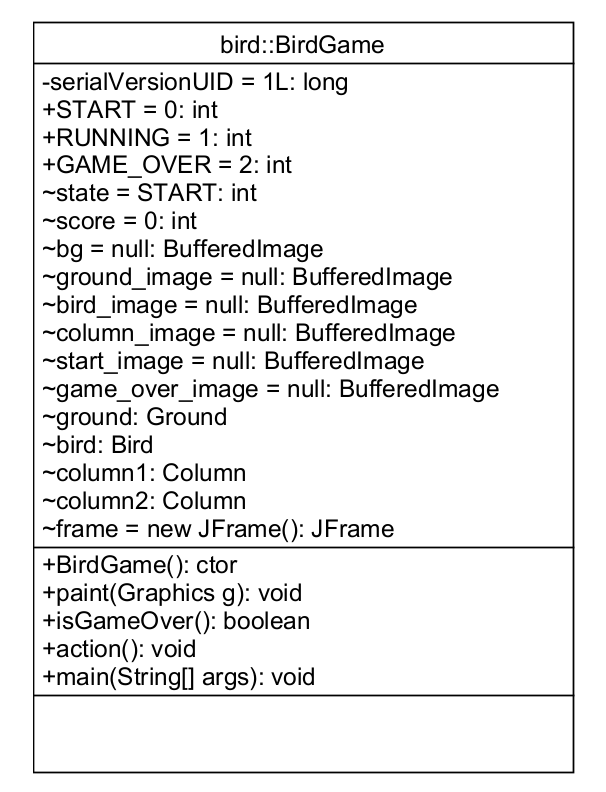
。

图2.1BirdGame类的程序类图

## 2.3 源程序及注释

package bird;

import javax.imageio.ImageIO;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.KeyAdapter;

import java.awt.event.KeyEvent;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.awt.image.BufferedImage;

import java.io.IOException;

// 游戏启动类

public class BirdGame extends JPanel {

private static final long serialVersionUID = 1L;

//定义游戏状态

public static final int START = 0; // 开始状态

public static final int RUNNING = 1; // 运行状态

public static final int GAME\_OVER = 2; // 结束状态

// 游戏的当前状态 默认0 - 开始状态

int state = START;

int score = 0; // 表示玩家的得分

static BufferedImage bg = null; // 背景图片

static BufferedImage ground\_image = null; // 地面图片

static BufferedImage bird\_image = null; // 小鸟图片

static BufferedImage column\_image = null; // 柱子图片

static BufferedImage start\_image = null; // 开始状态的图片

static BufferedImage game\_over\_image = null; // 结束状态的图片

// 静态代码块 一般用于加载静态资源(视频，音频，图片等)

static {

try {

bg = ImageIO.read(BirdGame.class.getResourceAsStream("bg.png"));

ground\_image = ImageIO.read(BirdGame.class.getResourceAsStream("ground.png"));

bird\_image = ImageIO.read(BirdGame.class.getResourceAsStream("0.png"));

column\_image = ImageIO.read(BirdGame.class.getResourceAsStream("column.png"));

start\_image = ImageIO.read(BirdGame.class.getResourceAsStream("start.png"));

game\_over\_image = ImageIO.read(BirdGame.class.getResourceAsStream("gameover.png"));

} catch (IOException e) {

e.printStackTrace();

}

}

Ground ground; //声明地面

Bird bird; //声明小鸟

Column column1 ; // 柱子1

Column column2 ; // 柱子2

// BirdGame的构造方法 (在new BirdGame() 的时候执行)

public BirdGame(){

ground = new Ground(); // 创建地面对象

bird = new Bird(); // 创建小鸟对象

column1 = new Column();

column2 = new Column();

// 柱子2的 x坐标 = 柱子1的坐标的基础上 + 244 （保证水平间距）

column2.x = column1.x + column1.distance;

}

public void paint(Graphics g) {

g.drawImage(bg , 0 , 0 , null);// 画背景

g.drawImage(bird.image , bird.x , bird.y , null);// 画小鸟

g.drawImage(column1.image , column1.x , column1.y , null);// 画柱子1

g.drawImage(column2.image , column2.x , column2.y , null);// 画柱子2

g.drawImage(ground.image , ground.x , ground.y , null);// 画地面

if (state == START){// 开始状态

g.drawImage(start\_image , 0 , 0 , null);

}

if (state == GAME\_OVER){// 结束状态

g.drawImage(game\_over\_image , 0 , 0 , null);

}

// 画分数

Font font = new Font("微软雅黑", Font.BOLD , 30); // 创建字体

g.setFont(font); // 给画笔设置字体

g.setColor(Color.BLACK); // 设置画笔颜色

g.drawString("分数：" + score , 30 , 50);

g.setColor(Color.WHITE); // 设置画笔颜色

g.drawString("分数：" + score , 28 , 48);

}

//检测游戏是否结束（实际上就是判断 小鸟与 地面或者柱子 是否相撞）

public boolean isGameOver(){

boolean isHit = bird.hitGround(ground) ||

bird.hitColumn(column1) ||

bird.hitColumn(column2);

return isHit ;

}

//游戏流程控制的方法

public void action() throws Exception {

frame.addKeyListener(new KeyAdapter() {

@Override

public void keyPressed(KeyEvent e) {

if (e.getKeyCode() == 32){ // 按下了 空格键

if (state == START){ // 如果是开始状态 单击 转换为运行状态

state = RUNNING;

}

if (state == RUNNING){ // 运行状态 单击 小鸟向上飞

bird.up(); //小鸟往上飞

}

if (state == GAME\_OVER){ // 结束状态 单击 转换为 开始状态

// 初始化游戏状态

bird = new Bird();

column1 = new Column();

column2 = new Column();

column2.x = column1.x + column1.distance;

score = 0; // 分数清零

state = START;

}

}

}

});

//给当前对象(画板)添加鼠标单击事件

this.addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {// 当单击鼠标的时候 执行该方法

if (state == START){ // 如果是开始状态 单击 转换为运行状态

state = RUNNING;

}

if (state == RUNNING){ // 运行状态 单击 小鸟向上飞

bird.up(); //小鸟往上飞

}

if (state == GAME\_OVER){ // 结束状态 单击 转换为 开始状态

// 初始化游戏状态

bird = new Bird();

column1 = new Column();

column2 = new Column();

column2.x = column1.x + column1.distance;

score = 0; // 分数清零

state = START;

}

}

});

// 死循环 ： {}内部的代码 会一直反复的执行

while (true){

if (state == START){

ground.step(); // 地面移动

bird.fly(); // 小鸟飞翔

}else if (state == RUNNING){ // 运行状态

ground.step(); // 地面移动

bird.fly(); // 小鸟飞翔

column1.step();// 柱子1 移动

column2.step();// 柱子2 移动

bird.down();// 小鸟移动

if (isGameOver() == true){ // 游戏结束

state = GAME\_OVER;

}

if (bird.x == column1.x + column1.width + 1 ||

bird.x == column2.x + column2.width + 1){

// 得分

score += 5;

}

}

repaint(); // 重画 即重新执行 paint方法

Thread.sleep(10);// 每10毫秒，让程序休眠一次

}

}

//创建画框对象（即：从java提供的资源中 获取 JFrame）

static JFrame frame = new JFrame();

// main方法 - 程序的入口（即：有main方法，程序才能运行）

public static void main(String[] args) throws Exception {

BirdGame game = new BirdGame();// 创建画板对象

frame.setSize(432,644);// 设置宽高

frame.setLocationRelativeTo(null); // 居中显示

// 设置关闭窗口 同时 使程序结束

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setVisible(true);// 设置可见性

frame.add(game); // 将画板放在画框上

//调用action

game.action();

}

}

**2.4 运行结果**



图2.2BirdGame类的运行结果图

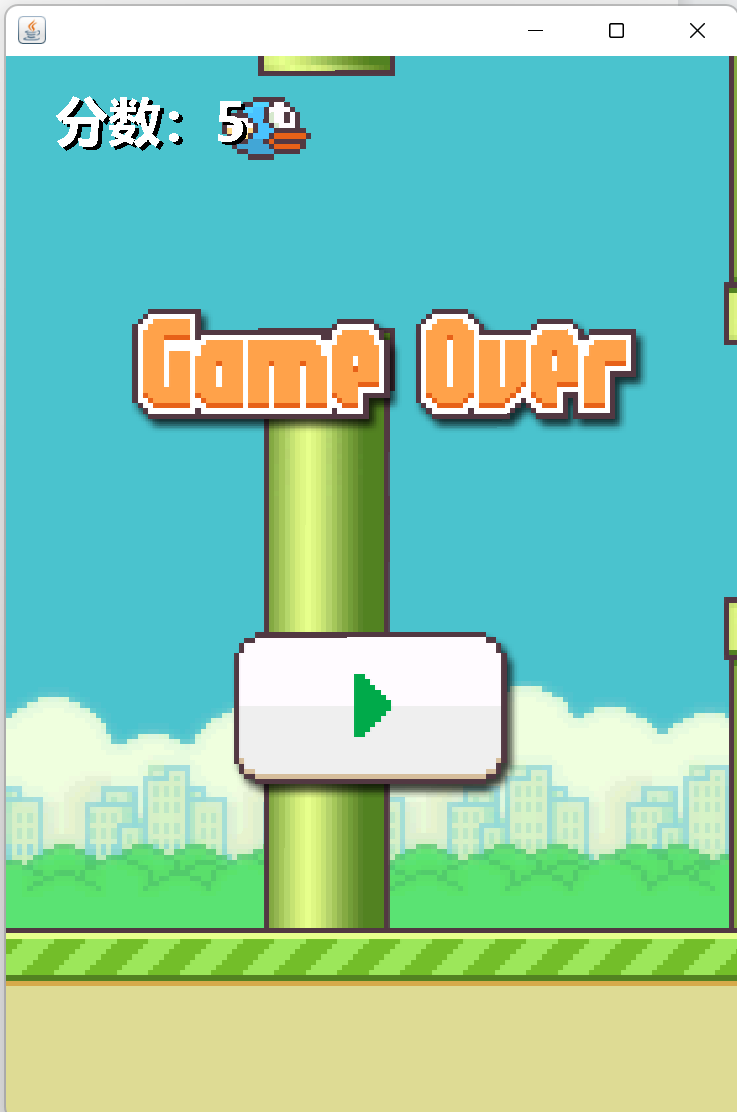


图2.3BirdGame类的运行结果图

# 3 总结

在本次实践课程中我学到了画框的基本使用和图片变量的处理，同时我了解到了JAVA面向对象编程思想的重要和基础概念，以及多种类的耦合关系。学习了Java中面向对象编程的基本概念和原则，包括类、对象、封装、继承、多态等。我掌握了基本的Java编程技能，能够独立开发简单的Java应用程序。同时，我也认识到Java的强大和广泛应用，为以后的学习和实践奠定了基础。Java 是一门很强大的编程语言，应用非常广泛。Java 不仅仅是用于开发 Java 应用程序，它还可以用于开发网络应用、Web 应用程序、移动设备应用程序、游戏等等。学习 Java 需要掌握基本的编程概念，如变量、循环、条件语句、面向对象编程等等。需要花费一定的时间去掌握这些基础知识。

# 参考文献

1. Kathy Sierra Bert Bates ．Head First Java（中文版）． 北京． 中国电力出版社． 2006
2. 耿祥义 . JAVA2实用教程 . 北京 . 清华大学出版社 . 2021

[3] 李刚著 . Java从入门到精通 . 北京 . 清华大学出版社 . 2016