



武汉大学

WUHAN UNIVERSITY

# 《JAVA 网络程序设计》

## 第十六小组 · 小鸡躲炸弹

### 实习报告

学 院: 遥感信息工程学院

班 级: 2010 班

学 号: 2020302131119

姓 名: 殷雨薇

指导教师: 王华敏

2021 年 6 月 11 日

## 一、实习内容

### 1. 本次实习目的与意义：

经过一学期的《Java 网络程序设计》课程的学习，我们小组完成了“小鸡躲炸弹”小游戏。该游戏的设计目的在于设计和开发一款方便广大用户使用的游戏，用于人们日常生活娱乐。在进行游戏设计开发的同时，也能提高我们的编程综合能力。

在本次实习中，我们通过课堂理论与实践相结合，深刻理解了 Java 的原理与特点，进一步掌握了程序设计的有关概念和开发方法，实际分析问题和解决问题的能力也得到了进一步提高。同时，在小组交流讨论的过程中，我的沟通合作技巧也得到了提高。

### 2. 功能设计：

核心功能：用键盘控制小鸡躲避炸弹。

所有功能：

- 游戏角色（小鸡）加载
- 炸弹在面板上随机乱序飞行
- 游戏玩家对小鸡的躲避操作
- 小鸡触碰到炸弹的爆炸效果显示
- 游戏结束时展现的时间显示

### 3. 小组分工：

李泽裕：进行需求分析，设计系统功能模块、GameUtil 工具类。

骆一川：设计系统界面（背景界面设计、小鸡形象设计、爆炸效果设计）、参数类（Constant 类）。

魏依新：设计小鸡类（Plane 类）、爆炸类（Explode 类）。

殷雨薇：设计方法判断小鸡与炸弹碰撞（GameObject 类）、炸弹类（Bomb 类）。

最后小组成员一起创建了 MyGameFrame 类，作为游戏主界面的承载对象。

## 二、系统功能需求分析

本游戏的需求是首先生成一个游戏界面，然后载入背景图和小鸡的图片，并

且设置小鸡由上下左右键控制其移动。在界面的 200\*200 像素处生成数十个炸弹，并让炸弹向随机方向移动，当碰到游戏界面边缘时反弹向另一个方向移动。游戏玩家通过上、下、左、右控制键控制小鸡向相应的方向移动。并且检测小鸡和炸弹的坐标是否有重合，如有重合则表示小鸡被炸弹炸到，则游戏结束。

系统的功能需求主要分为以下几类：

- 1. 游戏主界面需要呈现游戏背景、炸弹以及小鸡。
- 2. 游戏可以随机产生炸弹，炸弹能够按照随机产生的角度飞行，并能够在碰撞到面板边缘后自动返回。
- 3. 游戏中有一只小鸡，小鸡可以通过游戏玩家的操作进行上下左右的移动，来躲避随机飞行的炸弹。
- 4. 游戏需要能够监测到小鸡和炸弹是否相撞，如果没有相撞，则继续进行游戏。如果相撞，则游戏结束，显示游戏的时长。

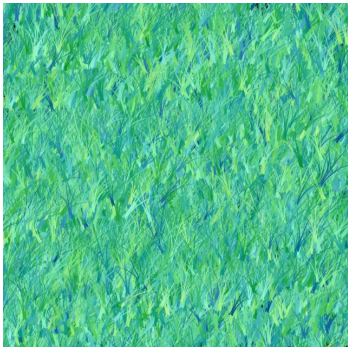
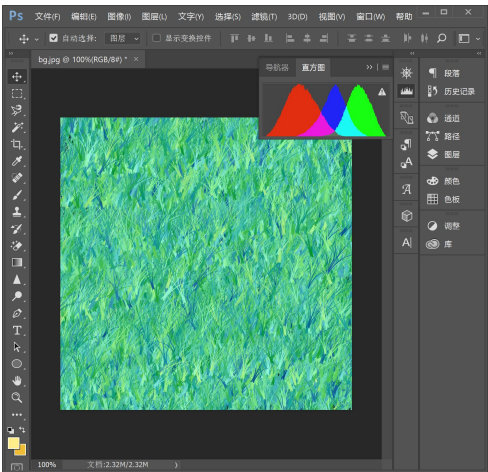
### 三、程序详细设计介绍

#### 1. 界面设计

##### 1.1游戏的背景界面设计

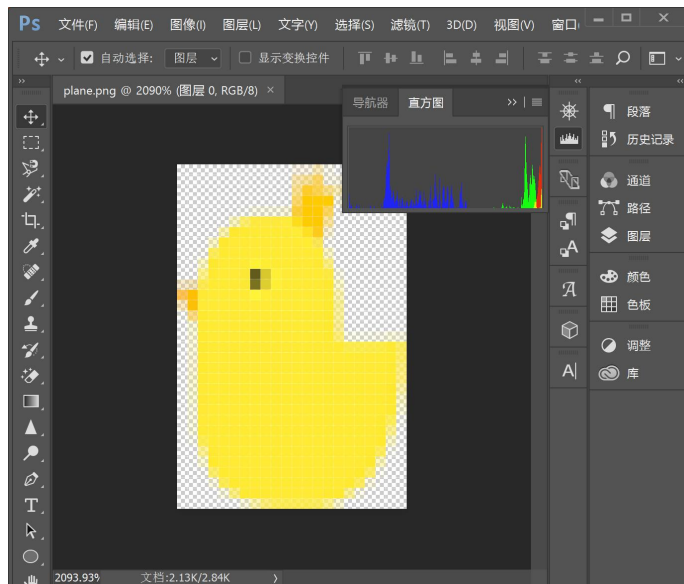
游戏的背景界面是在 Photoshop 中用草形草笔选择青色和蓝色画出，图像大小为 900\*900 像素，图片画好后存为 jpg 格式。

当游戏开始启动时，程序会自动读取背景图片，将背景图片展现在主面板之上。



## 1.2 游戏中的小鸡形象的设计

游戏中小鸡形象的设计主要使用了软件 PhotoShop，在空白面板上，先新建一个宽 22 像素、高 33 像素的画布，然后将画布的不透明度设置为 0%，再画出小鸡的图形，然后存为 png 格式。该形象的设计主要是为了方便后续小鸡与炸弹碰撞的检测需要。



## 1.3 爆炸效果的设计

游戏中在小鸡触碰到炸弹后，需要出现一个爆炸效果。

我们定义 Explode 类来表示爆炸的信息，它存储了一系列爆炸的图片，然后进行轮播。设计方法请老师详见骆一川同学的报告。

## 2. 程序设计

### 2.1 炸弹类

Bomb.java 定义了炸弹类，其中定义了炸弹的基本参数，包含 x 轴位置、y 轴位置、速度、宽度、高度等。

```

public class Bomb extends GameObject{

    double degree;
    public Bomb() {

        x = 200;
        y = 200;
        width=10;
        height = 10;
        speed = 3;
        degree = Math.random()*Math.PI*2;
    }
}

```

创建 draw 方法以绘制炸弹，并定义炸弹以任何角度飞行移动。

```

public void draw(Graphics g){
    Color c = g.getColor();
    g.setColor(Color.YELLOW);

    g.fillOval((int)x,(int) y, width, height);

    //炸弹沿着任意角度去飞
    x += speed*Math.cos(degree);
    y += speed*Math.sin(degree);

    if(x<0||x>Constant.GAME_WIDTH-width){
        degree = Math.PI - degree;
    }

    if(y<30||y>Constant.GAME_HEIGHT-height){
        degree = - degree;
    }

    g.setColor(c);
}

```

炸弹类我们用实心的黄色圆形实现。是在窗口固定位置（200,200）处生成炸弹，炸弹飞行的方向通过 Math.sin(degree) 定义，角度是随机的，速度定义为 3，并且遇见边界会反弹。

```

//炸弹沿着任意角度去飞
x += speed*Math.cos(degree);
y += speed*Math.sin(degree);

```

在定义炸弹的随机飞行时，我们调用了系统自带的 Math.random 函数，该函数可以随机生成数字，保证了炸弹飞行的随机性。

```

degree = Math.random()*Math.PI*2;

```

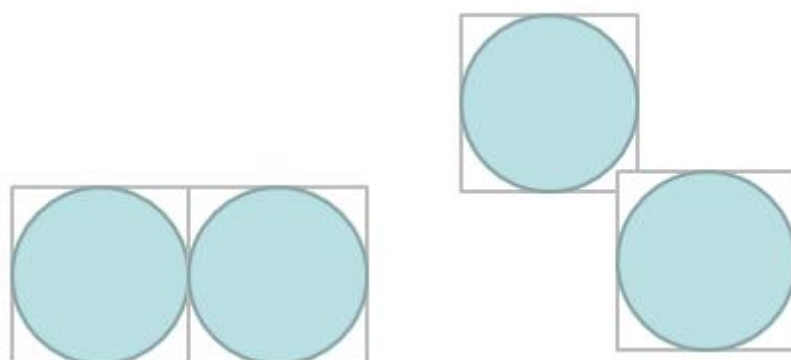
在炸弹类中，通过 draw(Graphics g) 绘制炸弹类，包含炸弹的颜色，炸弹实心圆形的绘制，并调用 Bomb 类中定义好的宽度，高度等属性。

主界面程序 MyGameFrame 中将创建一个 Bomb 类的新对象，并调用该对象类

中的各类参数，方法等，并重复产生若干个炸弹，完成炸弹类在游戏面板中的初始化。

## 2.2 矩形检测原理

在游戏中，多个元素是否碰在一起，实际上，通常是用“矩形检测”原理实现的。在该游戏中所有物体都可以抽象为“矩形”，我们只需要判断两个矩形是否相交即可。对于一些复杂的多边形和不规则物体，实际上是将他分解成了多个矩形，继续进行矩形检测。



Java 的 API 中，为我们提供了 Rectangle 类来表示矩形的相关信息，并且提供了 intersects() 方法，可以直接判断矩形是否相交。

```
// 小鸡和炸弹的接触检测！！  
boolean peng = Bombs[i].getRect().intersects(plane.getRect());
```

具体的实现方式是通过定义的类的 width、height 以及它的 x 坐标和 y 坐标实现。也就是说，在本游戏中所有物体包括小鸡和炸弹都有它们自己的矩形对象。圆形的定义也是有其长与宽。小鸡的图像定义虽然是小鸡形状，但是其只是绘制了小鸡形状的色彩，真实的图像保持了矩形形状，仍然包含长度与宽度。因此，当系统检测到它们的矩形元素有交集时，我们就判定小鸡和炸弹相撞，则游戏结束。

## 2.3 游戏类

该类定义了游戏对象以及对象的相关参数，包含游戏面板的坐标信息、图片加载信息、长度、宽度等。同时添加构造器，方便对象的调用。定义矩形检测类用于游戏结束的判定。

```

public class GameObject {
    Image img;
    double x,y;
    int speed;
    int width, height;

    public void drawSelf(Graphics g){
        g.drawImage(img, (int)x,(int) y, null);
    }

    public GameObject(Image img, double x, double y, int speed, int width, int height) {
        super();
        this.img = img;
        this.x = x;
        this.y = y;
        this.speed = speed;
        this.width = width;
        this.height = height;
    }

    public GameObject(Image img, double x, double y) {
        super();
        this.img = img;
        this.x = x;
        this.y = y;
    }

    public GameObject() {
    }

    /**
     * 返回物体所在的矩形。便于后续的接触检测
     * @return
     */
    public Rectangle getRect(){
        return new Rectangle((int)x, (int)y, width, height);
    }
}

```

## 2.3 游戏主界面的承载对象——MyGameFrame 类

在 Java 工程中,创建 MyGameFrame 的 java 文件作为游戏主界面的承载对象,包含游戏的主体面板,以及负责承载小鸡、炸弹等各类元素。例如,对小鸡和炸弹的初始化代码如下:

```

Plane plane = new Plane(planeImg,300,300);
Bomb[] Bombs=new Bomb[50];

```

该 Frame 通过调用和关联 Plane (小鸡类) 以及 Bomb (炸弹类), 并调用 Explode 类实现游戏结束时的爆炸效果。该游戏的开始是从 MyGameFrame 文件的启动开始, 而结束的判定主要依据的是炸弹和小鸡的位置信息。(如下图)



```

boolean peng = Bombs[i].getRect().intersects(plane.getRect());
if(peng){
    plane.live = false;
    if(bao ==null){
        bao = new Explode(plane.x, plane.y);

        endTime = new Date();
        period = (int)((endTime.getTime()-startTime.getTime())/1000);
    }
    bao.draw(g);
}

```

如果在游戏的过程中,有一个或多个炸弹的位置信息与小鸡目前的定位位置信息重复,则判定小鸡被炸弹炸到,游戏结束,展示 Game Over 页面。

其它类的介绍请老师详见其他小组成员的报告。

## 四、总结

这次的游戏设计和开发让我学到了很多東西:

1、增强了自身的动手能力,提升了实践能力。这个大作业让我们将理论知识与实际应用相结合,为今后解决实际问题打下基础。在完成大作业的过程中,我试着运用自己学到的理论知识去解决实际问题,从了解需求到最终完成游戏设计,在这期间经历的各种困难大大提高了我的实践能力。

2、本次游戏的设计和开发让我在学到新知识的同时对之前学习的知识进行了巩固。经过这段时间的学习,我对 Java 语言有更加深入的认识。我在完成作业的过程中也搜集了很多资料,让我了解和掌握了更多的相关知识,对本次游戏的设计和开发有了更加深刻的认识,使我在编写程序时的思路更加清晰,更加高效。

回顾这学期 JAVA 游戏的设计和开发,我现在还存在着许多不足,希望以后能带着学习的热情继续学习,努力将我在此课程中学习到的知识运用到实际生活中,为社会做出自己的贡献。最后,衷心感谢老师和同学们在此次设计开发中对我的帮助。