

基于 Java 的飞机大战游戏的设计与开发

杨海 段华琼

四川大学锦城学院 计算机与软件学院 四川 成都 611371

【摘要】飞机大战是一款经典的游戏，本文使用 Java 语言，基于 Eclipse2019 平台，设计开发了飞机大战游戏，实现了移动、射击、碰撞检测、积分记录等功能。

【关键词】Java; 飞机大战; 游戏设计

1 引言

随着手机、电脑等电子设备的不断普及，游戏已经逐渐变为人们生活娱乐中不可或缺的组成部分。Java 语言作为目前使用最广泛的编程语言之一，具有简单、面向对象、稳定、与平台无关、解释型、多线程、动态等特点^[1]。Java 最大的优势是一种表示层的应用技术，效率很高，容易学习^[2]。对于正在学习 Java 编程的人来说，飞机大战游戏的设计开发无疑是锻炼编程能力的良好途径。本文基于 Java 中的 Swing 用户界面库，设计开发了一个飞机大战游戏，详细阐述了游戏各个功能的实现方法和技术。

2 游戏总体设计

2.1 游戏功能简介

飞机大战是一款单人射击类游戏，游戏过程中玩家可以通过操纵英雄机发射子弹，敌机在碰撞到英雄机发射的子弹时被击毁，玩家得分；当英雄机触碰敌机或敌机子弹时，扣除英雄机生命值及火力值；当英雄机生命值为零时保存当前得分，游戏结束。

游戏的主要功能包括：

(1) 两类敌机：游戏中的敌机分为两类，分别为积分机和奖励机，积分机被击毁时玩家获得积分，奖励机被击毁时玩家获得奖励。

(2) 奖励模式：当奖励机被击毁时，玩家有二分之一几率获得生命值提升，有二分之一几率获得火力值提升。

(3) 惩罚模式：当英雄机碰撞到敌方子弹时，扣除一定生命值；当英雄机触碰到奖励机时，扣除一定量生命值与火力值；当英雄机碰撞到积分机时，扣除一定生命值与火力值并获得积分。

(4) 火力模式：为了使英雄机的攻击强度可以随

着操作技术发生变化，游戏设置火力模式。当英雄机火力值向上突破阈值时，英雄机攻击强度提升一个档次，最多可提升两个档次。提升档次后每次发射子弹会扣除火力值，向下突破阈值时，攻击强度下降一个档次，最低下降为初始状态。

(5) 难度递增模式：为了使玩家得分具有一定上限，防止游戏时间决定分数的情况出现，游戏设置了难度递增功能，游戏中的敌机强度会随时间增强（每 6 秒提升一次生命值）。这将需要玩家在有限的时间内通过更高的技术来获得更高的得分。

2.2 类设计

游戏共设计了 1 个抽象类，2 个接口、8 个普通类。

其中，FlyingObject 类是所有飞行物的父类，定义了抽象方法 step、outOfBounds，用于改变飞行物的坐标、判断飞行物是否出界；定义了默认方法 shootBy，用于判断敌人是否被子弹击中。Award 接口和 Enemy 接口分别是奖励机接口和敌机接口，分别定义了抽象方法 getType 和 getScore，用于获取奖励类型及获取得分。普通类包括 Score 类、Hero 类、AirPlane 类、Plane 类、ShootGame 类，以及 Bullet 类等子弹类。其中 Hero 类、AirPlane 类、Plane 类、Bullt 类分别定义了英雄机、敌机（积分机、奖励机）、子弹的属性，比如英雄机的子弹发射速度与数量、英雄机生命值、敌机的移动速度、出现频率、两种类型敌机的数量比、子弹的速度等。Score 类为得分类，用于存储当前积分及获取历史最高积分，ShootGame 类为主类，带有主函数。

类之间的逻辑结构如下图所示：

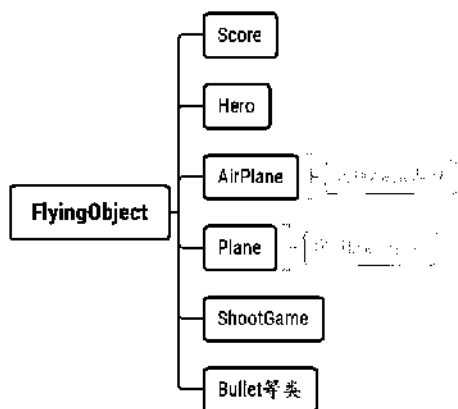


图 1 类结构图

3 游戏功能的实现

3.1 初始化游戏场景

首先在主类中创建 JFrame 类的窗口对象 frame，然后创建主类的面板对象 game，将 game 面板添加到 frame 窗口中。为了界面的美观，设置窗口宽为 530，高为 815，且使窗口在最上层居中显示可见，最后设置窗口默认关闭操作（关闭窗口时退出游戏），窗口左上角坐标为（0,0）。

3.2 飞行物的相关操作

为了实现对飞行物的多种操作，对不同类型的飞行物分别创建了不同的类，每个类均继承 FlyingObject 类。在 FlyingObject 类中定义了 BufferedImage 类变量，用来储存图片；定义了整型变量储存图片的宽高及坐标；定义了移动、出界判断、碰撞检测等抽象方法。

（1）初始化飞行物

在主类中使用静态代码块来缓存图片，以便于其它类实例化对象时调用图片。定义 FlyingObject 类型的数组 flyings 用于储存敌机和敌机子弹的对象，敌机及敌机子弹类对象将会自动向上转型。创建无返回值的 action 方法，在此方法中创建鼠标侦听器对象，同时重写鼠标的移动和点击事件，获取当前鼠标的坐标为英雄机坐标，点击鼠标任意键以更改游戏运行状态（开始状态改为启动状态、结束状态改为启动状态），对鼠标是否出界进行判定，重写鼠标移出、移入方法。当鼠标移出时，游戏从启动状态改为暂停状态；当鼠标移入时，游戏从暂停状态改为启动状态。创建定时器对象，重写 run 方法，将飞行物移动、射击、碰撞检测、删除越界飞行物以及游戏状态判断等方法添加到 run 方法内，每 10 毫秒执行一次 run 方法，重写 paint 方法，将 repaint 方法也添加到 run 方法的方法体中，并对方法体进行异常捕获。定时器代码如下：

```
Timer timer=new Timer();// 创建定时器对象
int interval=10;// 时间间隔 (以毫秒为单位)
timer.schedule(new TimerTask(){
// 按规定时间间隔运行 run 方法
    public void run(){// 每 10 毫秒运行一次
        try{
            if(state==RUNNING){// 运行状态时执行
                /* 此处代码省略(包含飞行物移动、射击、碰撞检测、删除越界飞行物 以及游戏状态判断等方法) */
            }
            repaint();// 调用 paint() 方法
        }catch(Exception e){
        }
    }
},interval,interval);// 定时计划
}
```

（2）增添飞行物

对于敌机类飞行物，创建返回值为 FlyingObject 类的方法 nextOne。利用 Java 自带的随机数类库生成随机数对象，对随机数的大小进行判定，根据判定结果返回积分机对象或奖励机对象。由于所有飞行物都继承自唯一的抽象类，为了尽量减少变量的数量，创建敌机数组用于储存所有的敌机对象和所有的敌机子弹对象，创建子弹数组，用于储存所有的英雄机子弹对象。创建无返回值的用于获取敌机对象的方法，调用 nextOne 方法生成对象，再调用 Arrays.copyOf 方法对敌机数组进行扩容，同时将生成的敌机对象储存在敌机数组最后的位置上。对于敌机或英雄机子弹对象的生成同理，创建无返回值的生成敌机子弹或生成英雄机子弹的方法生成敌机子弹对象或英雄机子弹对象，用同样的方法储存在相应的数组中。英雄机子弹的生成方法如下：

```
Bullet[]bs=hero.shoot();// 获取子弹对象（数组的长度随火力值变化）
bullets=Arrays.copyOf(bullets,bullets.length+bs.length);// 扩容
System.arraycopy(bs,0,bullets,bullets.length-bs.length,bs.length);
// 数组的追加
```

（3）删除飞行物

当敌机、子弹出界或与英雄机发生碰撞时，应将数组中对应位置的对象删除。为了避免空间的浪费，将待删除对象与数组中最后一个对象交换位置，同时利用 Arrays.copyOf 方法对数组进行缩容，以达到删除对象的目的。在删除被击毁的敌机对象时，利用 instanceof 关

键字对敌机的类型进行判断, 如果被击毁的是积分机, 则将对象向下转型为积分机类, 触发积分机对象中获取积分的方法获得积分; 如果被击毁的是奖励机, 则将对象向下转型为奖励机类, 触发奖励接口中获取奖励类型的方法, 获取奖励类型, 根据奖励类型调用增加火力值的方法或增加生命值的方法获得火力值或生命值奖励。

(4) 飞行物出界判定及碰撞检测

以面板左上角为坐标原点, 向右延伸为 x 轴正半轴, 向下延伸为 y 轴正半轴。对于子弹对象, 子弹对象的坐标为图像中心坐标, 由于子弹的图片较小, 不计子弹图片的宽和高。对于英雄机子弹, 子弹向上运动, 当子弹的 y 轴坐标小于零时判定为出界; 对于敌机子弹, 子弹向下运动, 当子弹的 y 轴坐标大于窗口的高时判定为出界。对于敌机对象, 当敌机上边界的 y 轴坐标大于窗口的高时判定为出界。

对于敌机和英雄机, 分别在各自的类中定义碰撞检测方法。用其图片的宽与高定义一个矩形域, 当敌机对象与英雄机对象的矩形域重叠时, 则代表发生了碰撞。对于子弹的碰撞检测方法, 当子弹的坐标处于矩形域内时, 则代表子弹击中了飞机。最后将两种检测方法添加到定时器 `run` 方法的方法体中, 这样所有飞行物每进行一次移动便会进行一次碰撞检测。

3.3 记录得分

用文本文档保存游戏得分, 以行为单位记录每次的得分, 通过 IO 输出流保存每次的得分。游戏开始时, 通过 IO 输入流读取文档, 将历史最高得分显示在面板上, 同时面板上还会依次显示当前得分、生命值与火力值。

这些直观的数据可以帮助玩家拥有更好的游戏体验, 达到更高的分数。

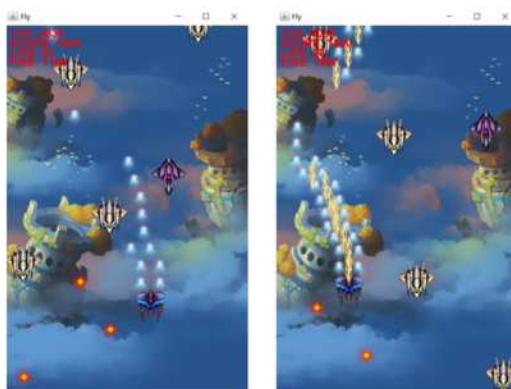


图 2 游戏界面

4 结语

飞机大战是一款非常经典的游戏。本文基于 Java 面向对象的编程思想, 在 Eclipse 平台下完成了一个飞机大战游戏的开发, 通过设计多个功能模块实现了移动、射击、碰撞检测、积分记录等基本功能。经测试, 游戏可以正常运行使用。由于水平有限, 游戏的设计实现过程仍有待进一步改进。

【参考文献】

- [1] Cay S. Horstmann. Java 核心技术·卷 I, 基础知识 [M]. 北京: 机械工业出版社, 2016.
- [2] 孙佳美. 基于 Java 飞机大战的设计与实现 [J]. 无线互联科技, 2016, No. 77: 59-60.