

**《程序设计方法学》课程设计报告**

**题 目：飞机大战**

**成 员：**

软件工程1703 17408070814 曹庆超

软件工程1703 17408070828 焦宇珠

软件工程1703 17408070726 高洋洋

**二O一九年七月**

目录

**[1课程设计目的及意义](#_Toc13460_WPSOffice_Level1)** **[2](#_Toc13460_WPSOffice_Level1)**

**[2设计内容](#_Toc15711_WPSOffice_Level1)** **[2](#_Toc15711_WPSOffice_Level1)**

**[3需求分析](#_Toc851_WPSOffice_Level1)** **[3](#_Toc851_WPSOffice_Level1)**

[3.1功能需求](#_Toc10127_WPSOffice_Level1) [3](#_Toc10127_WPSOffice_Level1)

[3.2数据需求](#_Toc28544_WPSOffice_Level1) [3](#_Toc28544_WPSOffice_Level1)

[3.3其它需求](#_Toc29933_WPSOffice_Level1) [4](#_Toc29933_WPSOffice_Level1)

**[4 系统设计](#_Toc15711_WPSOffice_Level2)** **[4](#_Toc15711_WPSOffice_Level2)**

[4.1系统功能设计](#_Toc15711_WPSOffice_Level3) [4](#_Toc15711_WPSOffice_Level3)

[4.2类及模式设计](#_Toc851_WPSOffice_Level3) [5](#_Toc851_WPSOffice_Level3)

[4.3数据库设计](#_Toc10127_WPSOffice_Level3) [6](#_Toc10127_WPSOffice_Level3)

[4.4主要算法程序流程图](#_Toc28544_WPSOffice_Level3) [8](#_Toc28544_WPSOffice_Level3)

**[5 系统实现及优化](#_Toc851_WPSOffice_Level2)** **[9](#_Toc851_WPSOffice_Level2)**

[5.1主要功能实现及调试](#_Toc29933_WPSOffice_Level3) [9](#_Toc29933_WPSOffice_Level3)

[5.3 优化方案](#_Toc6996_WPSOffice_Level3) [12](#_Toc6996_WPSOffice_Level3)

[5.4 实现效果及测试代码](#_Toc22729_WPSOffice_Level3) [12](#_Toc22729_WPSOffice_Level3)

**[6 总结](#_Toc10127_WPSOffice_Level2)** **[15](#_Toc10127_WPSOffice_Level2)**

**1课程设计目的及意义**

程序设计方法学是讨论[程序](https://zh.wikipedia.org/wiki/%E7%A8%8B%E5%BA%8F" \o "程序)的性质以及[程序设计](https://zh.wikipedia.org/wiki/%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1" \o "程序设计)的理论和方法的一门学科，是研究和构造程序的过程的学问，是研究关于问题的分析，环境的模拟，概念的获取，需求定义的描述，以及把这种描述变换细化和编码成机器可以接受的表示的一般的方法。本次课程设计立足于程序设计方法学，编写java语言飞机大战程序。通过本次课程设计，能够熟练应用多种程序设计方法，深入理解Java面对对象编程思想，掌握MySQL数据库以及Eclips等开发工具使用，将所学的程序世纪方法、java程序设计和数据库, 通过一个实际项目的分析、设计与实现进行综合的运用，培养学生分析、解决实际问题的能力以及软件开发能力。

**2设计内容**

飞机大战基于java swing 可视化窗口实现，游戏运行时，界面显示当前英雄机生命值和得分值，英雄机由鼠标移动控制，当鼠标点击时，游戏开始。鼠标移出窗口时游戏暂停，界面显示暂停提示。当英雄机生命耗尽时，游戏结束，界面显示结束提示。

游戏设计角色分别为英雄机，子弹，其中游戏实现了英雄机的鼠标键盘双重控制，当窗口顶端敌机随机出现向下飘落时，由英雄机自动发出的子弹击中实现敌机生命减少，当敌机或Boss子弹与英雄机碰撞时，其自身消失，英雄机减少生命数。当英雄机子弹击落小蜜蜂时，获得随机奖励类型分别为增加生命和增加子弹射出个数。

**3需求分析**

**3.1功能需求**

1. 英雄机在用户的控制下可以自由移动，但不可以移动到窗体外。英雄机可以连续发射子弹，子弹的起始位置和英雄机位置有关。
2. 敌机在窗口顶部随机产生，产生后垂直向下移动。
3. 小蜜蜂在窗口顶部随机产生，产生后向下向右移动，当碰到窗口边界时向向相反方向移动。
4. Boss可以发射子弹，每个Boss在一个发射时刻可以发射两枚子弹。Boss在窗口顶部随机产生，产生后以随机速度左右移动。。
5. 英雄机子弹由战机产生，垂直向上飞行，并且可以击中敌机和敌机子弹
6. 英雄机击落敌机会获得一定分数，当分数高于50分时英雄机获得大型飞翼子弹，不会因碰撞其他飞行物消失，提升英雄机战斗力。
7. 英雄机击落小蜜蜂时，随机获得加生命或提升火力奖励
8. Boss每50秒出现一次，击落后，英雄机得分加1000。

**3.2数据需求**

英雄机：英雄机位置，英雄机生命值，英雄机伤害值

敌机：敌机位置，敌机生命值，敌机生成速度，敌机移动速度

英雄机子弹：子弹位置，子弹发射速率，子弹移动速度，子弹伤害值

BOSS子弹：子弹位置，子弹生成速率。子弹移动速度，子弹伤害值

Bee 道具：道具类型，道具位置，小蜜蜂移动速度

Boss: Boss位置，Boss生命值，Boss伤害

**3.3其它需求**

天空背景的移动

天空背景的加速移动

英雄机控制的两种方式

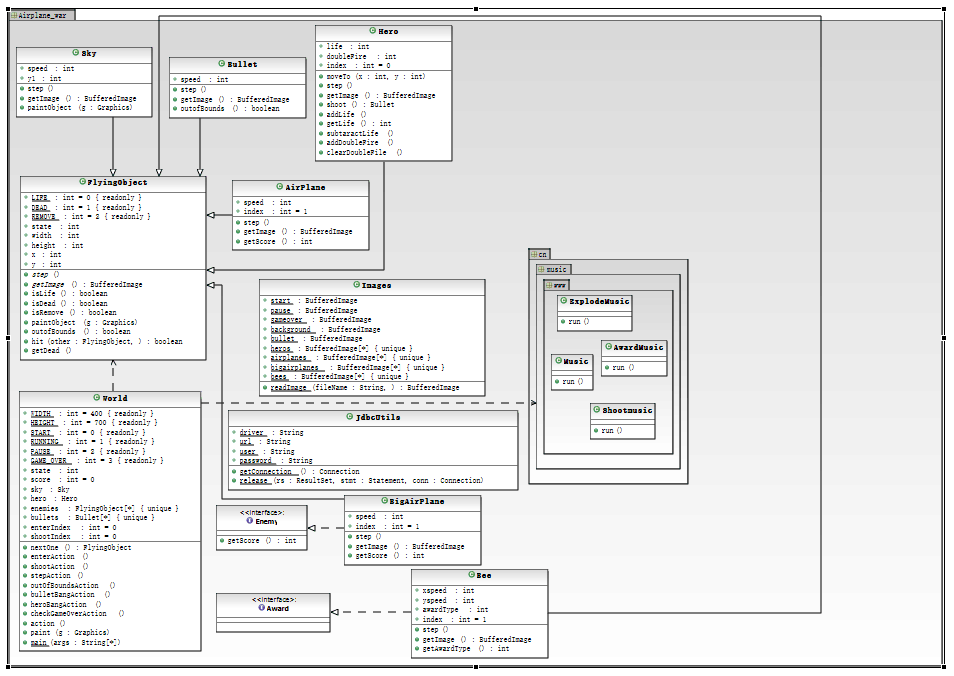
**4 系统设计**

**4.1系统功能设计**

玩家管理：绘制玩家   
 按键管理：操作控制（鼠标移动和键盘操作）   
 地图管理：绘制地图、地图移动   
 子弹管理：绘制子弹、子弹移动   
 奖励管理：奖励处理   
 爆炸管理：绘制爆炸、移除爆炸   
 BOSS管理：绘制BOSS、BOSS移动   
 资源管理：敌机与英雄机的越界删除  
 业务逻辑处理：产生多个敌机、多个子弹、碰撞、

**4.2类及模式设计**

（1）.类图设计



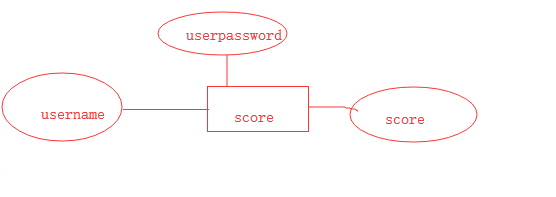
(2).设计模式应用

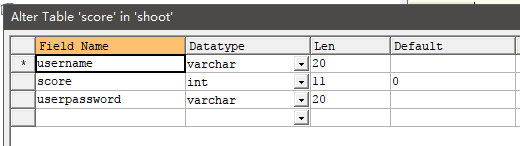
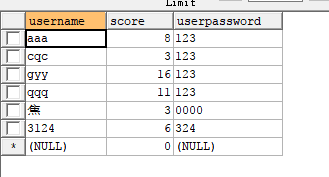
<1>.装饰模式：可以在不改变一个对象本身功能的基础上给对象增加额外的新行 为，该程序实现了两个接口，增加了得分和奖励功能！

<2>.观察者模式：本次程序设计采用Swing实现窗口功能可视化，用键盘实现英雄记操控，两者均为观察者模式，响应动作监听窗口事件和鼠标键盘事件。

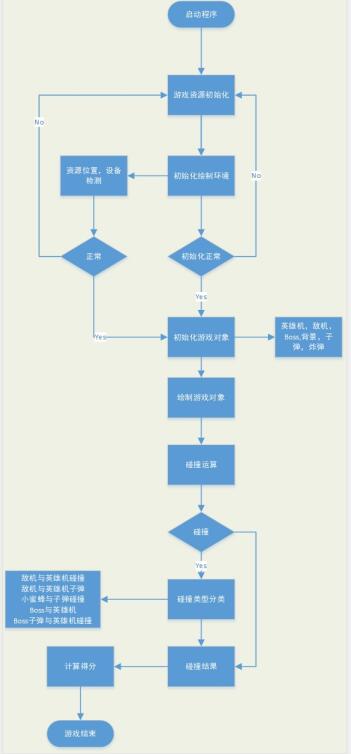
**4.3数据库设计**

（1）.数据库设计E-R图



1. .数据库关系模式
2. .数据库表设计

**4.4主要算法程序流程图**



**5 系统实现及优化**

**5.1主要功能实现及调试**

1. .英雄机的颜色变化，英雄机的鼠标控制移动和键盘操作

<实现>：英雄机两张不同图片在Image类中用数组存放，Hero类中重写getImage方法，在主类定时器中调用paintObject方法，每十秒 重绘一次，实现两张图片的循环切换。鼠标控制即在主类定时器run()方法中根据英雄机存活状态重写mouseMoved()等Mouse 操作函数。

1. .英雄机子弹跟随英雄机位置变化生成，根据得分实现英雄机 发射较大子弹的“大招”

<实现>：英雄机随鼠标移动，在英雄机Hero类里添加shoot()方法，让不同火力值的（即一次射出子弹个数）子弹对象构造函数参数为英雄机坐标经计算后的值。为实现大招，新建类BigSkill 继承Bullet类，以上述同样方法在Hero类 中编写bigSkill方法，当得分大于50分时，在World类中调用该方法，即可实现大招

1. .敌机在窗口顶部随机位置生成并实现下落，小蜜蜂奖励类型 的随机实现

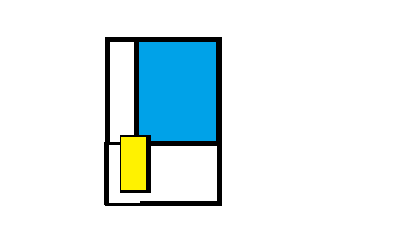
<实现>：敌机初始横坐标有随机数生成，生成后纵坐标每10ms向下移动step 像素，实现显示向下飘落，小蜜蜂实现Award接口，由随机数产生奖励类型。

1. .为敌机和BOSS添加生命值，为BOSS增加发射子弹和随机速 度左右移动功能

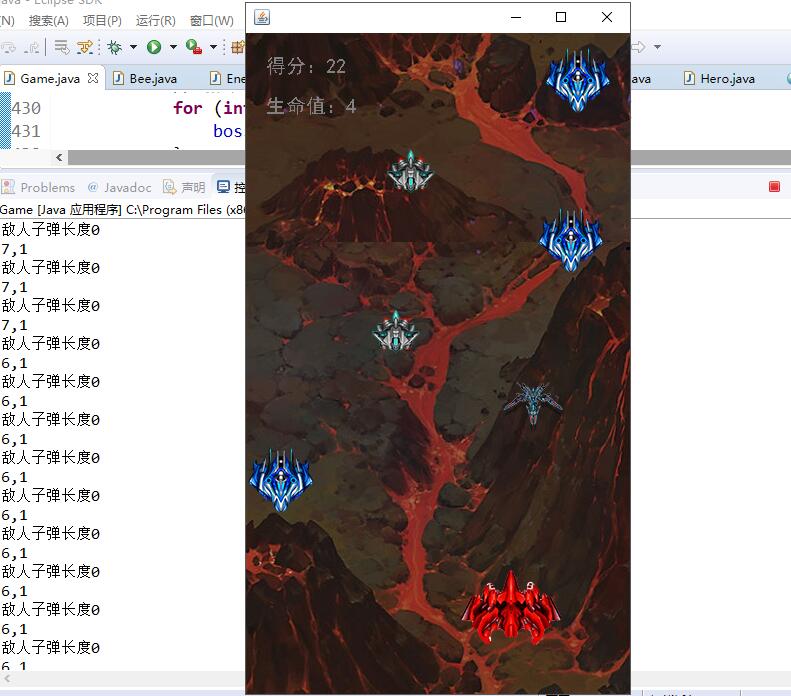
<实现>：实现原理同英雄机，即在父类构造函数中申明初始生命，在Boss 类中编写BossShoot()用与英雄机射击同样的方法实现发出子弹

1. .实现敌机与英雄机,英雄机子弹与敌机，Boss子弹与英雄机， Boss子弹与英雄机子弹之间的碰撞作用

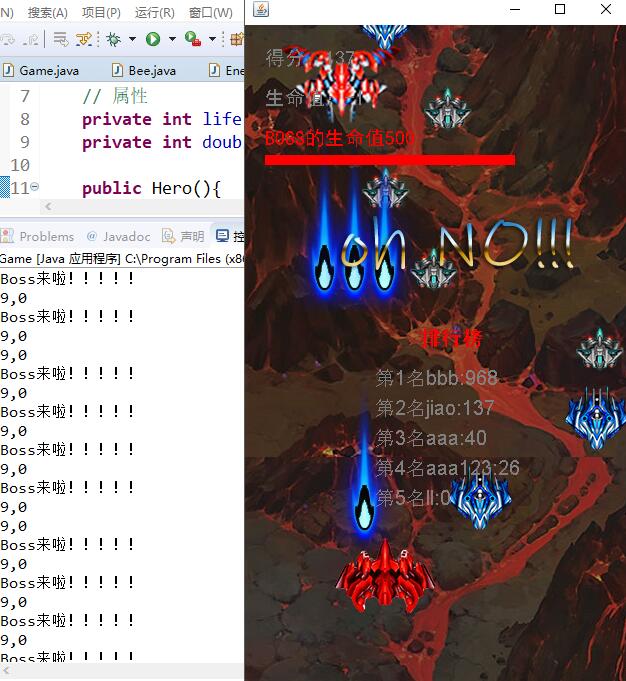
<实现>：

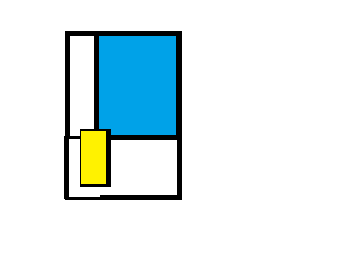
1. .连接数据库实现分数排序

调试截图：  
（1）当前子弹存活数量



1. Boss 来袭测试



**5.2 技术难点与解决方案**

Boss子弹添加和Boss状态判断后产生

由于world类只产生了一个Boss对象，因此在第二次出现时会产生问题，此时将状态用分支结构判断此时的状态，将第一个Boss初态给第二个，以此类推解决问题。

以下是解决代码：

**public** FlyingObject bossNextOne() {

Boss b = **new** Boss();

b.state = FlyingObject.***LIFE***;

boss = b;

**return** boss;

}

**5.3 优化方案**

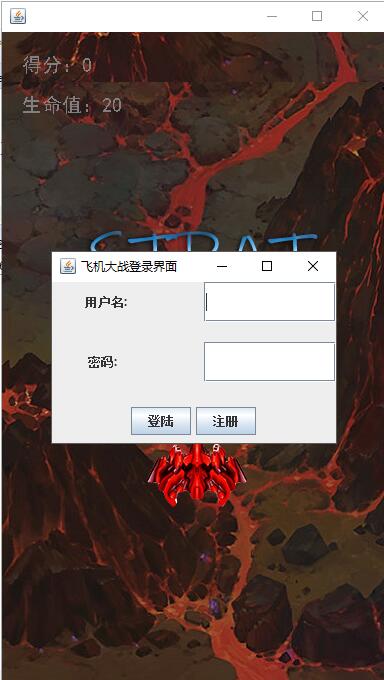
处理越界对象时多次缩容代码繁杂，采用一次缩容可以简化代码

敌机天空Bee均可写自己的独立代码，但为了提高代码的复用性，采用继承并调用super方法实现

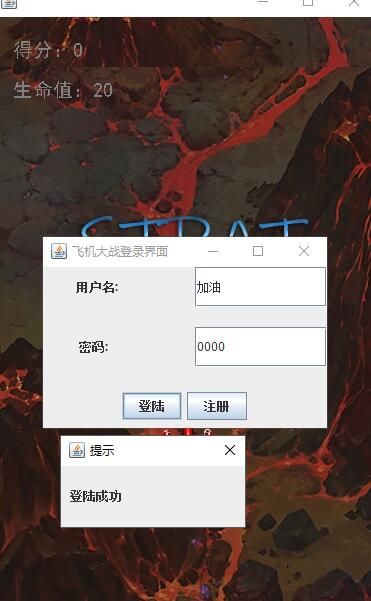
在基础功能实现的基础上，添加BOSS和键盘操作，在游戏开始时链接数据库实现用户的登陆注册功能。

**5.4 实现效果及测试代码**

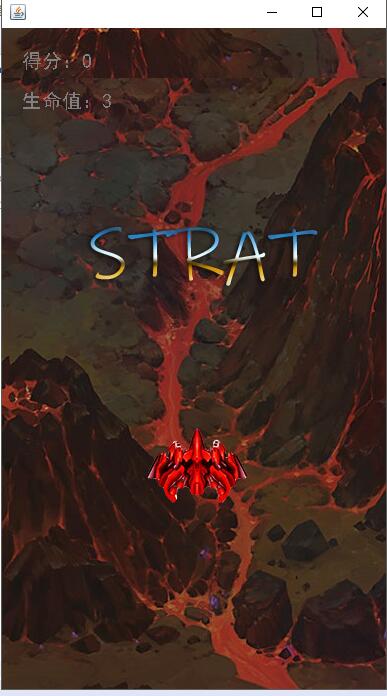
（1）.游戏开始登陆注册



注册登录成功



**（2）.游戏开始及暂停**





1. Boss出现及英雄机死亡排行榜，boss血条绘制



**6 总结**

本次课程设计是程序设计方法学，java面向对象编程，数据库设计三者的综合使用及实践，提高了我们对已学知识的综合实践应用能力，深化理解面对对象编程思想。通过编写类和函数较多的 实践案例能够真正提升我们对一个项目的认识高度，有助于未来对其他项目的开发。

就业指导课程详细 全面介绍了当前就业大环境以及我们当前所需学习的知识，有助于我们尽早地做好职业生涯规划，权衡利弊，现在最应该做的事情是努力提升自己的技术水平。为未来的就业等做好充足准备。

参考文献

【1】Java 编程思想 机械工业出版社 （美）Bruce Eckle

【2】java 从入门到精通 清华大学出版社 明日科技

【3】. 巧用Jsp和Java连接Mysql数据库[J]. 现代商贸工业 李君