



关于实验课

1. 使用腾讯课堂上课，如遇到技术故障将改用腾讯会议；
2. 为方便考勤，请同学们将昵称改成“学号-真实姓名”；
3. 上课不定时发起签到，请同学们不要迟到早退。



哈爾濱工業大學(深圳)
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

規格嚴格



功夫到家

1920 — 2017

面向对象的软件构造导论

实验一：飞机大战功能分析

2022春

哈尔滨工业大学（深圳）



目录

01

本学期实验总体安排

02

实验一说明

03

作业提交



目录

01

本学期实验总体安排

02

实验一说明

03

作业提交



我们的目标

The screenshot displays an IDE environment with the following components:

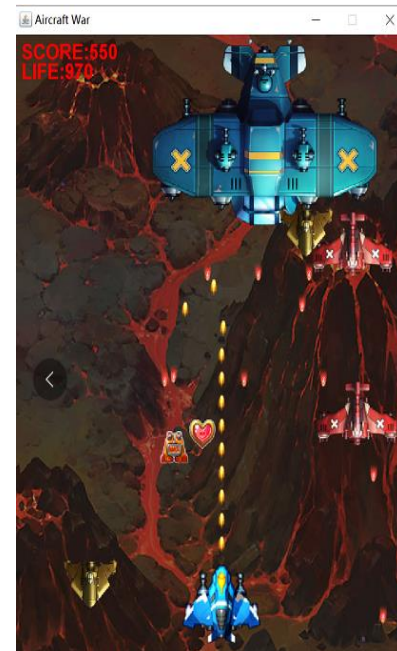
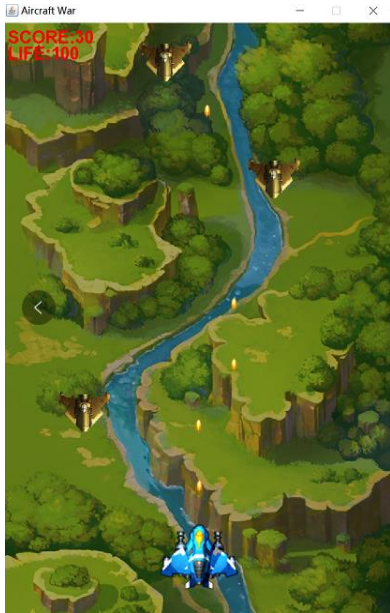
- Project Explorer:** Shows the project structure for 'AircraftWar-base', including files like 'bgm_boss.wav', 'bomb_explosion.wav', 'bullet.wav', 'bullet_hit.wav', 'game_over.wav', 'get_supply.wav', and a 'test' directory containing various UML files.
- Structure:** Lists the classes and methods within the 'MediumGame' package, such as 'MediumGame()' and 'tick(): void !BaseGame'.
- Run Console:** Displays the output of the program, showing 'Hello Aircraft War'.
- Game Interface:** A central window titled 'Aircraft War' featuring three difficulty mode buttons: '简单模式' (Simple Mode), '普通模式' (Normal Mode), and '困难模式' (Hard Mode). Below these buttons is a '音效' (Sound Effect) toggle set to '开' (On).
- Code Editor:** Shows the 'MediumGame.java' file with tabs for 'EasyGame.java' and 'BaseGame.java'.
- System Tray:** Includes a system clock showing 17:30 on 2022/2/7, a temperature indicator at 16°C, and a taskbar with various application icons.

游戏难度分为简单、普通、困难三个等级



I

本学期实验总体安排



- 游戏主界面
- 英雄机、普通敌机
- 飞机移动
- 英雄机子弹发射
- 碰撞检测

模板程序

- 加上其他类型的敌机
- 加上3种道具

实验一

- 英雄机改成单例模式
- 敌机和道具改成工厂模式

实验二

使用Junit进行单元测试

实验三

- 采用策略模式实现子弹的不同弹道发射
- 采用数据访问对象模式实现排行榜

实验四

- 添加游戏难度选择
- 添加排行榜界面
- 多线程实现音效的控制
- 多线程实现火力道具

实验五

- 用观察者模式实现炸弹道具生效
- 采用模板模式实现三种游戏难度

实验六

本学期实验总体安排

实验项目	一	二	三	四	五	六
学时数	2	2	2	4	2	4
实验内容	飞机大战 功能分析	单例模式 工厂模式	Junit与单元测试	策略模式、 数据访问 对象模式	Swing 多线程	模板模式 观察者模式
分数	4	6	4	6	6	14
提交内容	UML类图、 代码	UML类图、 代码	单元测试 代码	UML类图、 代码	代码	项目代码、 实验报告

实验课程共**16**个学时，**6**个实验项目，总成绩为**40**分。



目录

01

本学期实验总体安排

02

实验一说明

03

作业提交



实验目的

- ◆ 深入理解面向对象的基本思想；
- ◆ 结合实例，深入理解面向对象分析和设计的方法；
- ◆ 掌握UML类图的绘制方法。



实验任务

1. 分析飞机大战系统功能;
2. 导入飞机大战模板程序;
3. 使用PlantUML插件绘制敌机和道具类的类图及继承关系;
4. 在模板程序基础上, 增加精英敌机和道具类的代码。

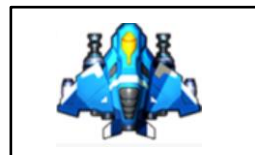


实验步骤

1 飞机大战系统功能分析

角色设定

● 英雄机



● 敌机



● 道具



● 子弹





实验步骤

1 飞机大战系统功能分析



游戏规则

- 游戏难度：简单、普通、困难
- 子弹发射、弹道变化
- 道具生成、获取和使用
- 计算生命值和得分、总分排行榜

请参考实验指导书！



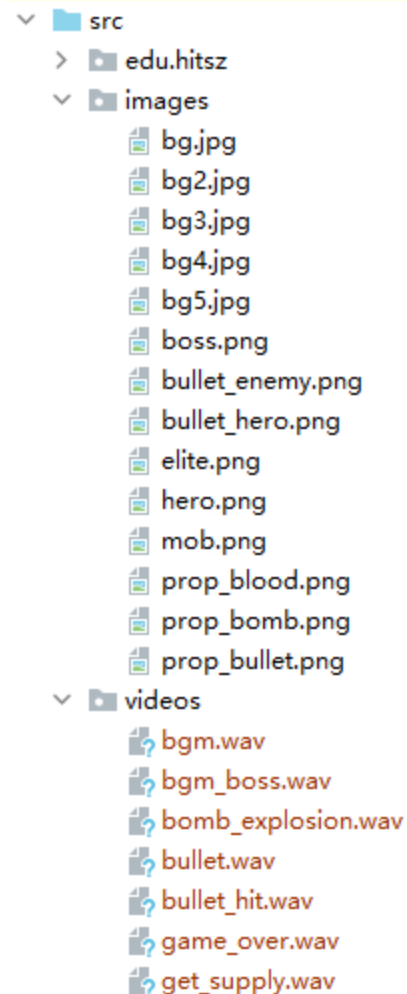
实验步骤

1 飞机大战系统功能分析

界面和音效

游戏地图

音效开启和关闭



请参考实验指导书!



实验步骤

2

模板程序导入

本实验提供了飞机大战的模板代码 **AircraftWar-base.zip**, 已实现如下功能:

- 游戏主界面
- 英雄机、普通敌机
- 飞机移动
- 英雄机子弹发射
- 碰撞检测
- 统计并显示得分和英雄机生命值

安装Java集成开发环境**IntelliJ IDEA**后可导入该项目运行。

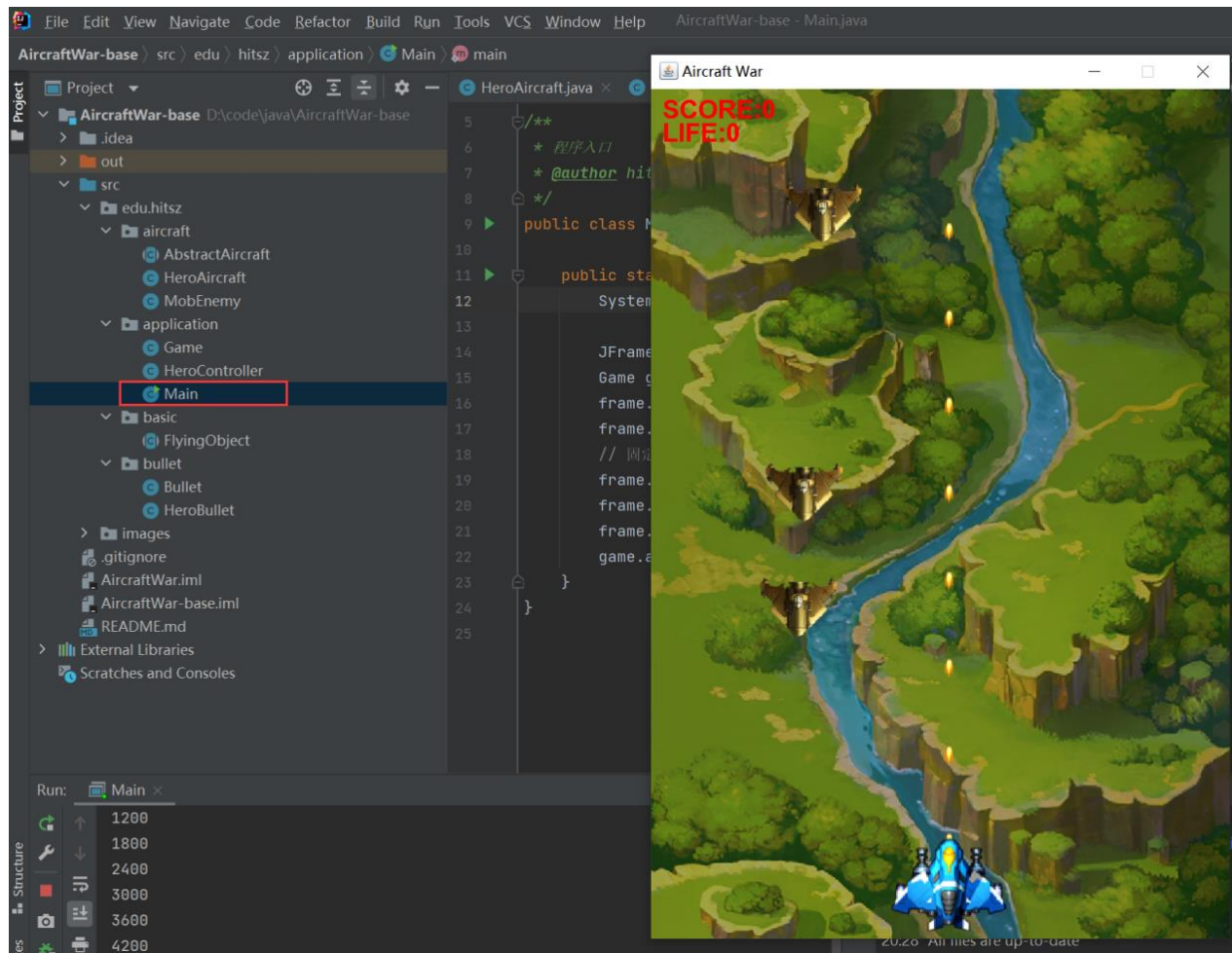


实验步骤

2

模板程序导入

- 安装IntelliJ IDEA
- 安装JDK11
- 导入项目
- 运行 Main.java

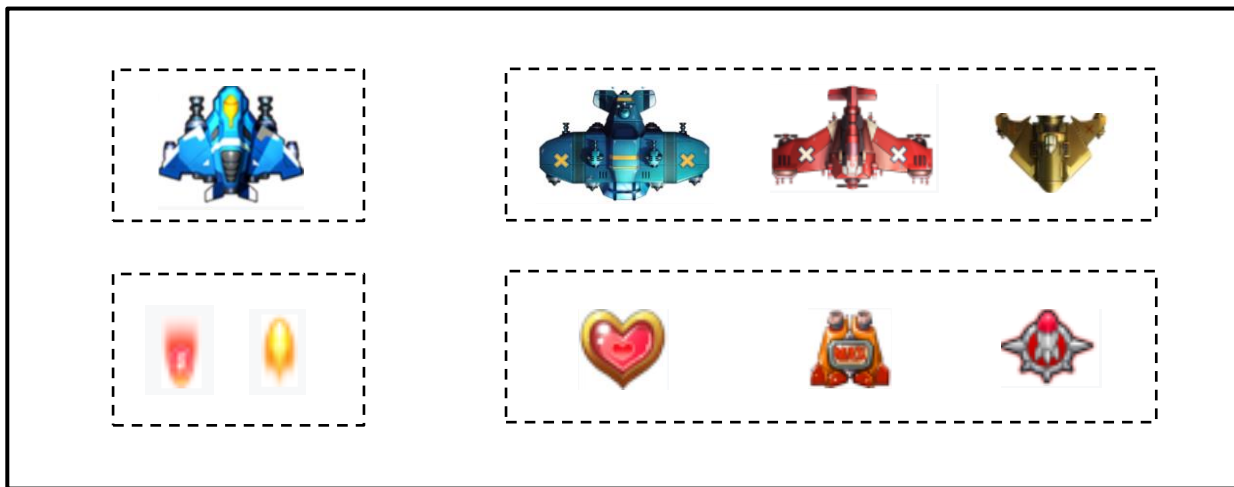




实验步骤

3 绘制UML类图

根据面向对象设计原则，分析和设计游戏中的所有敌机和道具类，并使用PlantUML插件绘制相应的UML类图及继承关系。类图中需包括英雄机、所有敌机、道具及它们所继承的父类。



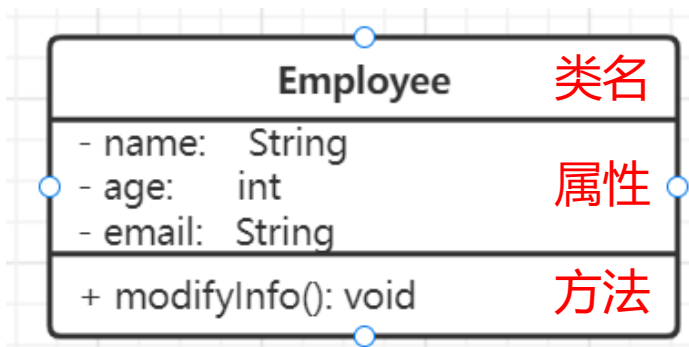


实验步骤

3

绘制UML类图

类图 (Class Diagram) : 用来显示系统中的类、接口以及它们之间的静态结构和关系的一种静态模型。



```
public class Employee {  
    private String name;  
    private int age;  
    private String email;  
  
    public void modifyInfo() {  
        .....  
    }  
}
```



实验步骤

3

绘制UML类图

类与类之间的关系: 泛化、实现、依赖和关联。
其中关联又分为**一般关联**和**聚合关系**, **组合关系**。

01 ↑

泛化关系

继承关系, 子类继承父类的所有行为和属性。如: 老虎和动物

02 ↑

实现关系

类与接口的关系, 表示类是接口所有特征和行为的实现者。如: 鸟和飞行。

03 ↓

依赖关系

一种使用关系, 一个类的实现需要其他类的协助。如: 驾驶员和汽车。

04 ↓

一般关联

对象之间的一种引用关系, 用于表示一类对象与另一类对象之间的联系。如: 顾客和商品。

05 ↓

聚合关系

整体与部分的关系, 且部分可以离开整体而单独存在。如: 汽车和轮胎。

06 ↓

组合关系

整体与部分的关系, 但部分不能离开整体而单独存在。如: 公司和部门。

实验步骤

3

绘制UML类图

The screenshot shows the IntelliJ IDEA interface with the 'AbstractAircraft' class selected in the project structure. A context menu is open, and the 'Diagrams' option is highlighted. The 'Show Diagram...' option is selected, which will generate a UML class diagram for the selected class and its subclasses.

UML Class Diagram Structure:

- FlyingObject** (Base Class)
 - Attributes: locationX (int), locationY (int), speedX (int), speedY (int), image (BufferedImage), width (int), height (int), isValid (boolean)
 - Operations: FlyingObject(), FlyingObject(int, int, int), forward(), crash(FlyingObject) (boolean), getLocationX() (int), getLocationY() (int), setLocation(double, double) (void), getSpeedY() (int), getImage() (BufferedImage), getWidth() (int), getHeight() (int), notValid() (boolean), vanish() (void)
- AbstractAircraft** (Subclass)
 - Attributes: maxHp (int), hp (int)
 - Operations: AbstractAircraft(int, int, int, int), decreaseHp(int) (void), getHp() (int), shoot() List<AbstractBullet>
- HeroAircraft** (Subclass)
 - Attributes: shootNum (int), power (int), direction (int)
 - Operations: HeroAircraft(int, int, int, int), forward() (void), shoot() List<AbstractBullet>
- MobEnemy** (Subclass)
 - Operations: MobEnemy(int, int, int, int), forward() (void), shoot() List<AbstractBullet>

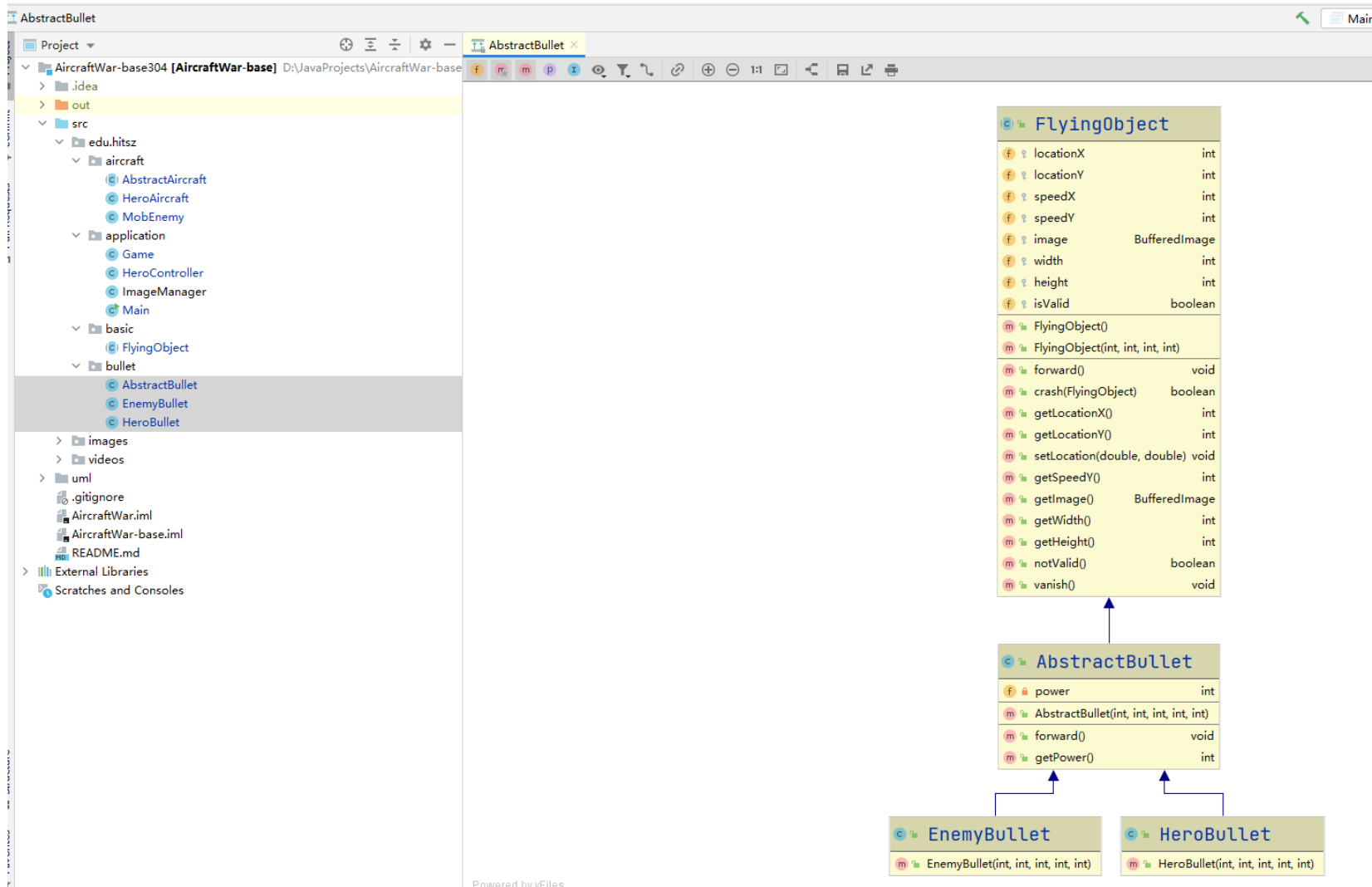
这是什么关系?

- A、泛化
- B、实现
- C、依赖
- D、关联

实验步骤

3

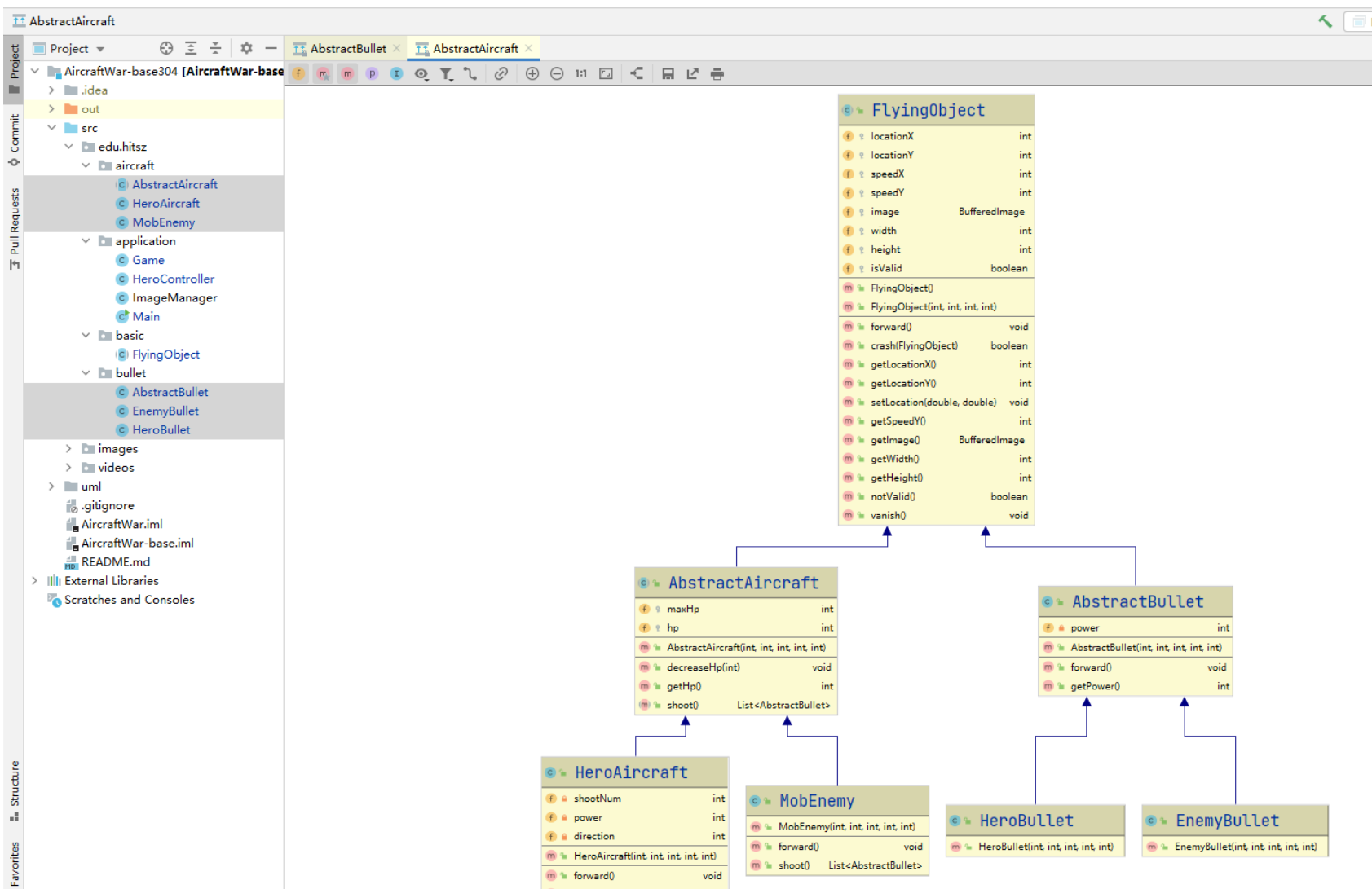
绘制UML类图



实验步骤

3

绘制UML类图





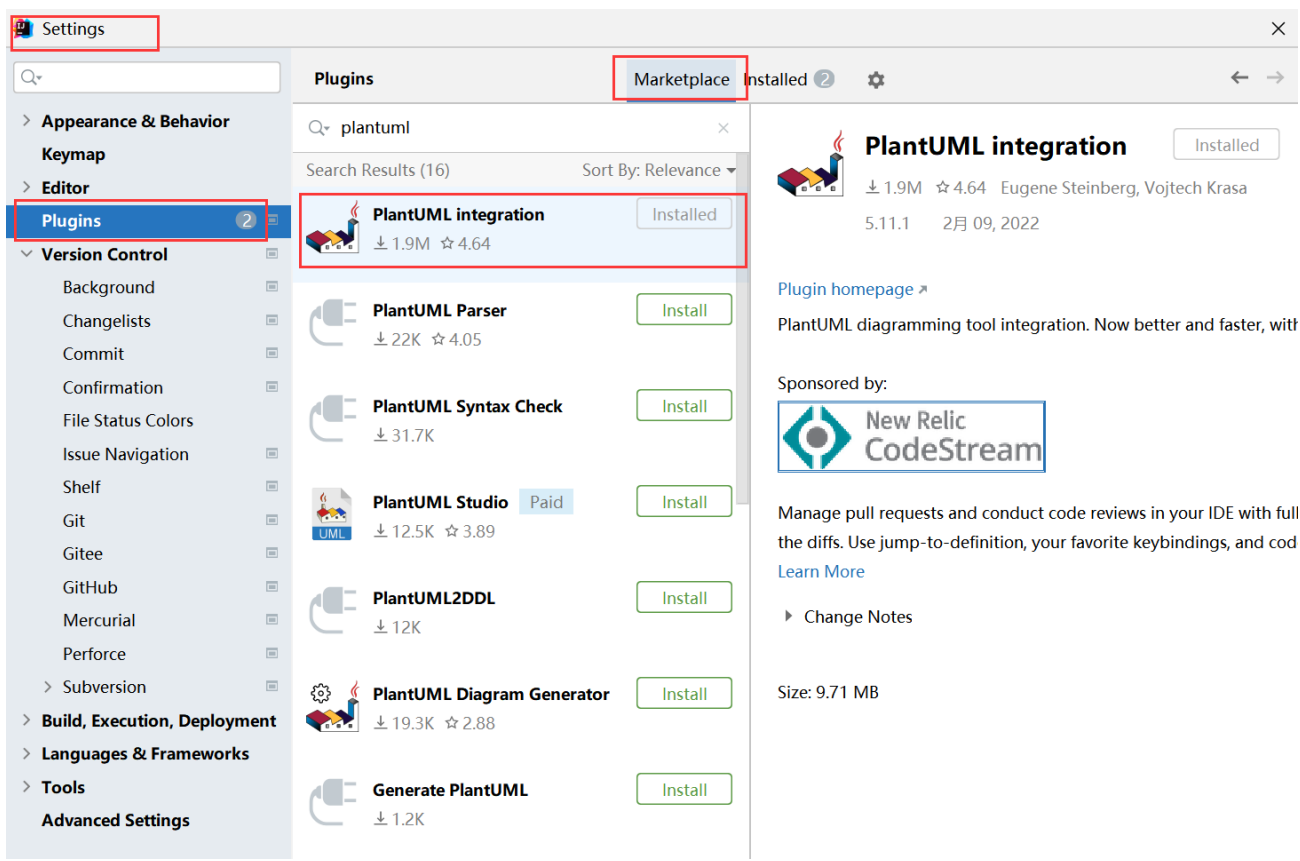
实验步骤

3

绘制UML类图

选用IntelliJ IDEA的**PlantUML**插件

PlantUML支持快速绘制类图、时序图、用例图等UML图



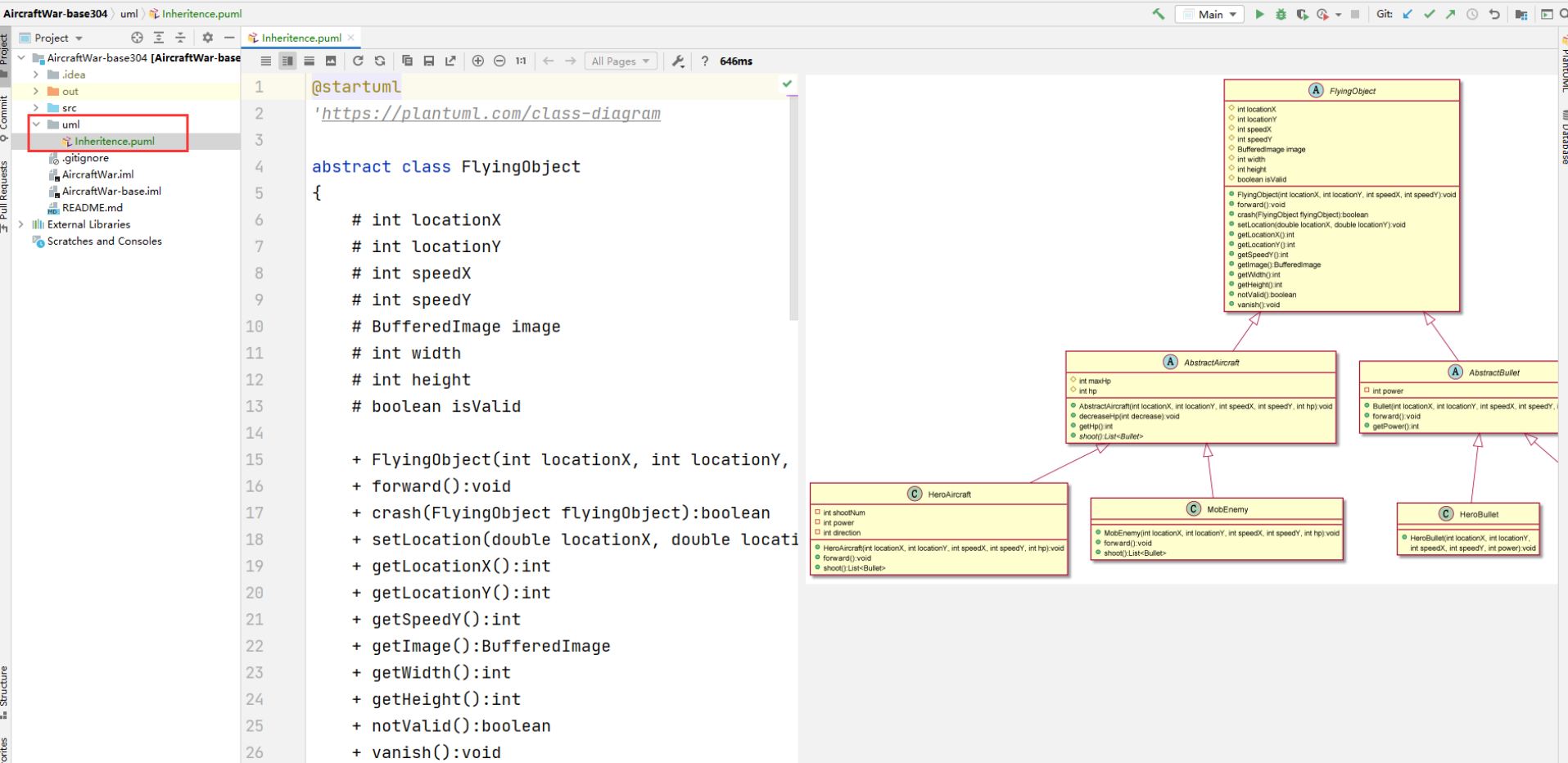


实验步骤

3

绘制UML类图

PlantUML类图示例





实验步骤

3

绘制UML类图

PlantUML类图的语法和功能:

请参考官网说明: [类图的语法和功能 \(plantuml.com\)](http://plantuml.com)



类之间的关系是用以下符号定义的。

类型	符号	绘图
扩展	< --	
组成	*--	
聚合	o--	

可以用 .. 来代替 -- , 这样就可以有一条虚线。



实验步骤

4

编写代码

根据设计的UML类图，在模板代码基础上编写代码，添加相应的敌机类和道具类。

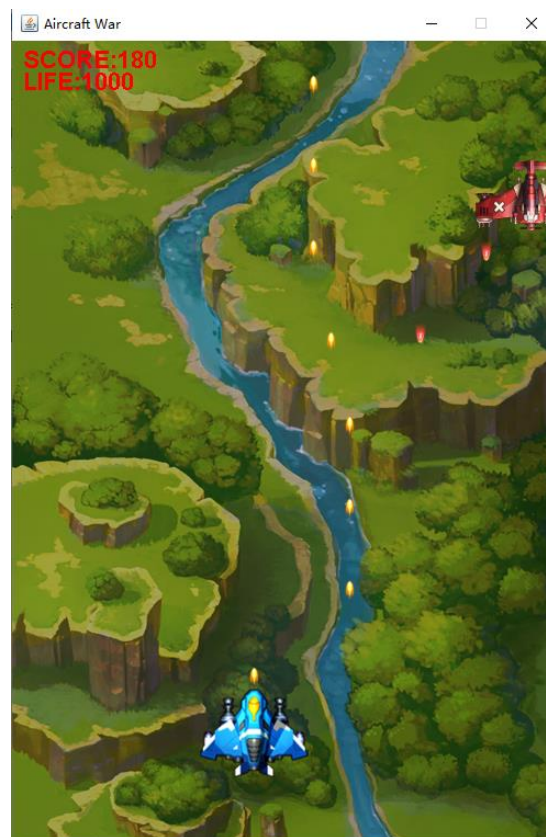
本实验可暂不实现Boss机，或者建一个Boss类，具体功能先不实现。





本次实验的目标（1）

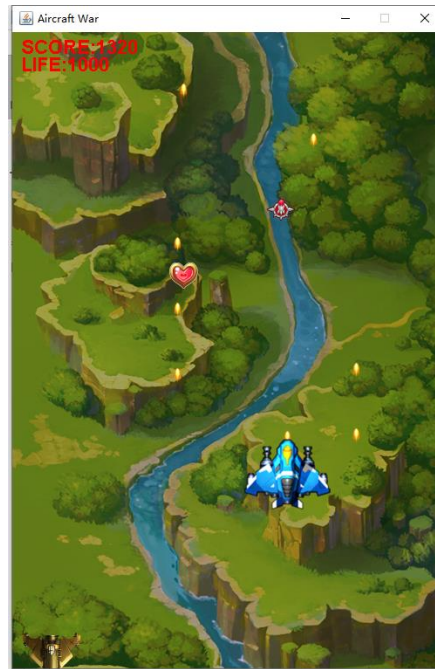
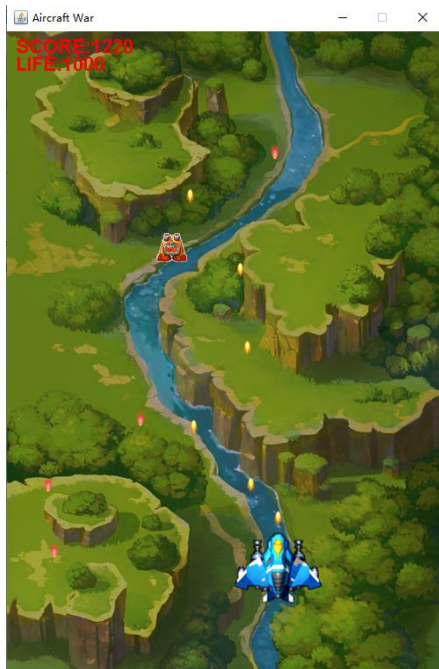
- ✓ 每隔一定周期随机产生一架普通敌机或精英敌机;
- ✓ 精英敌机按设定周期发射子弹;





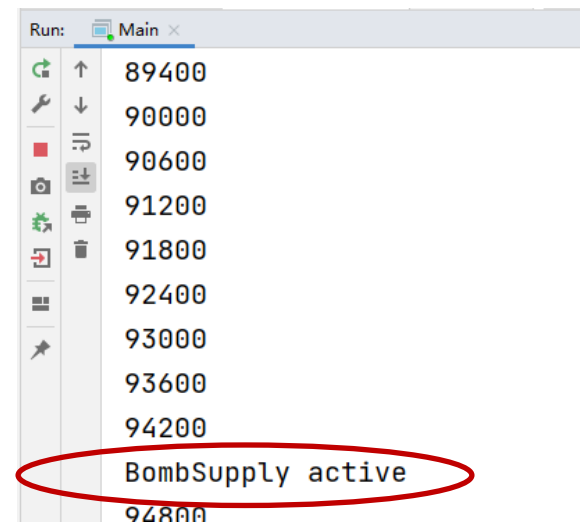
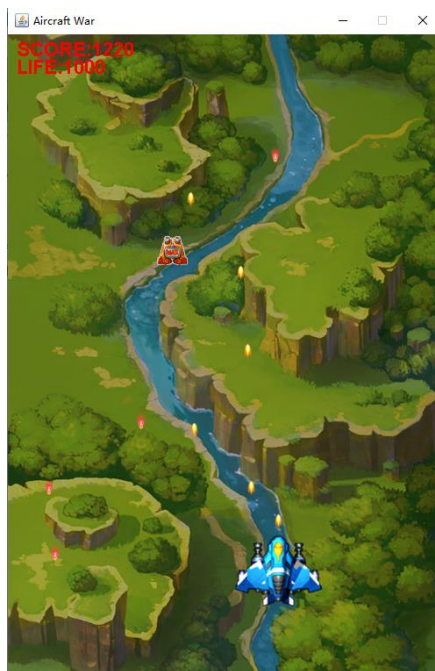
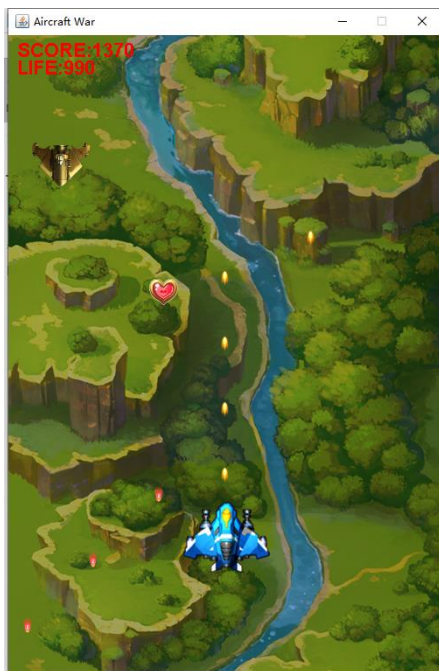
本次实验的目标（2）

- ✓ 精英敌机坠毁后随机产生某种道具（也可能不产生道具）；



本次实验的目标

- ✓ 英雄机碰撞道具后，道具自动生效。其中加血道具生效时可使英雄机恢复一定血量，火力道具和炸弹道具生效时只需在控制台打印“FireSupply active!” “BombSupply active!” 语句即可。





目录

01 本学期实验总体安排

02 实验一说明

03 作业提交



作业提交

- **提交内容：**项目代码

包括：

- ① 使用PlantUML插件绘制的UML类图及继承关系；
- ② 正常运行的代码。

- **截止时间**

实验课后一周内提交至**HITsz Grader** 作业提交平台，具体截止日期参考平台发布。

- 登录网址：：<http://grader.tery.top:8000/#/login>
- 推荐浏览器：Chrome
- 初始用户名、密码均为学号，登录后请修改

➤ 上传后可自行下载以确认是否正确提交



**同学们
请开始实验吧！**