

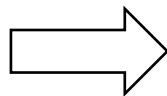
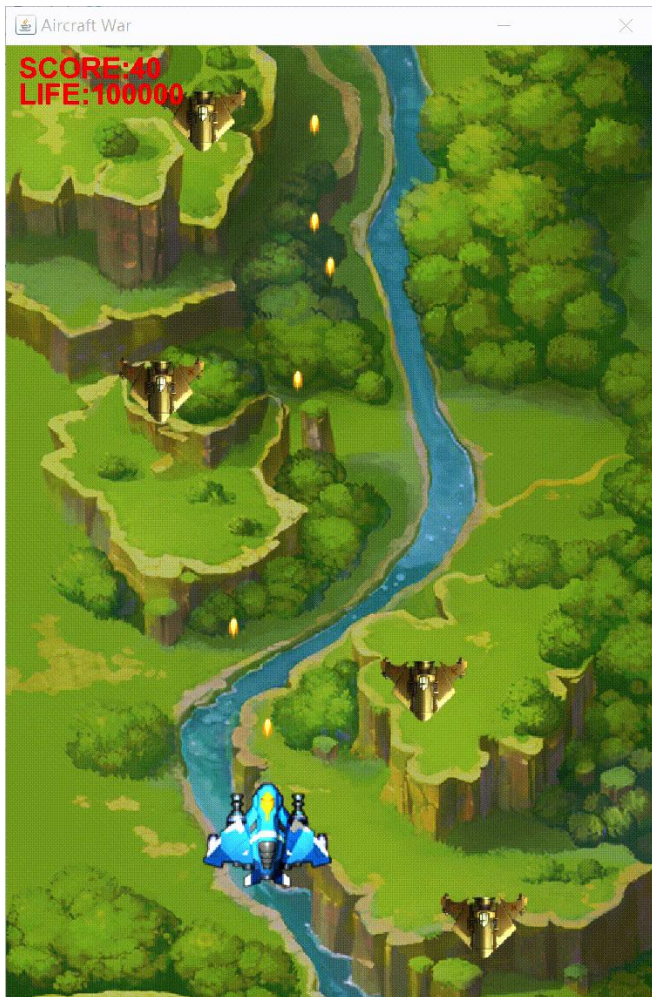


实验一：飞机大战功能分析

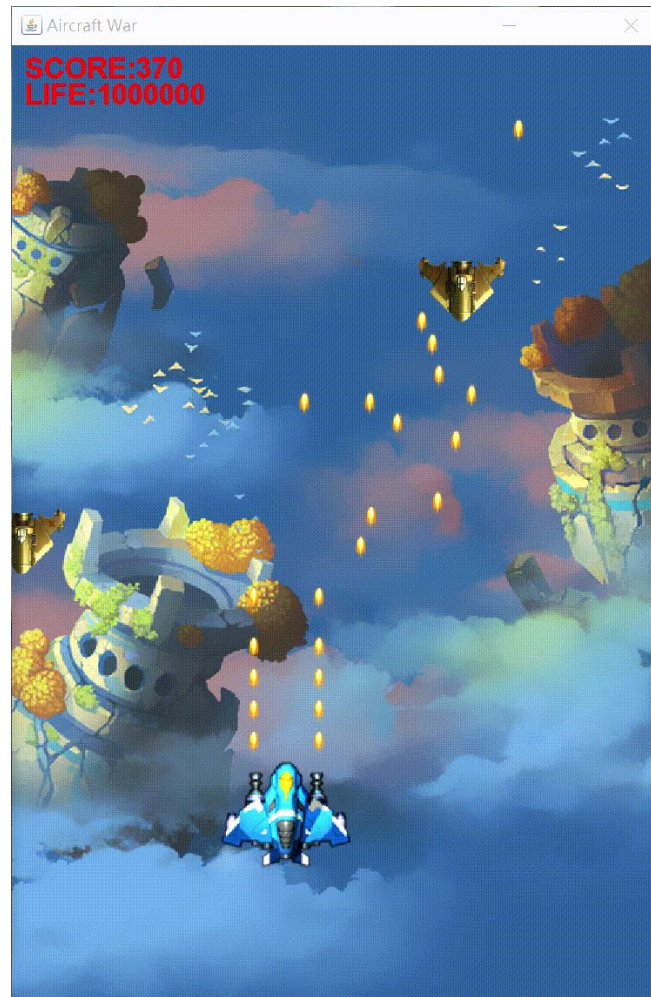
实验与创新实践教育中心 · 计算机与数据技术实验教学部

本学期实验总体安排

我们的
开始

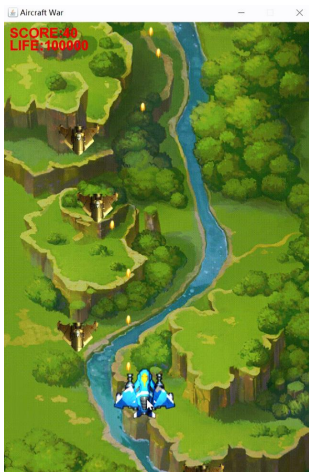


我们的
目标

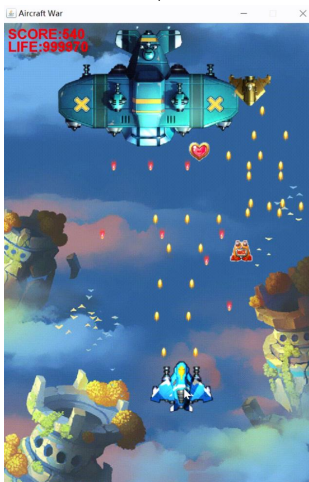


本学期实验总体安排

初始版本



最终版本



游戏主界面
英雄机移动
英雄机子弹直射
碰撞检测
统计得分和生命值

重构代码，采用**单例模式**
创建英雄机
重构代码，采用**工厂模式**
创建敌机和道具

重构代码，采用**策略模式**
实现不同弹道发射
采用**数据访问对象模式**
实现得分排行榜

采用**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

初始版本

01

绘制UML类图

创建精英敌机并直射子弹
精英敌机随机掉落三种道具
加血道具生效



02

03

添加**JUnit单元测试**
创建Boss和其它敌机

04

05

使用**Swing**添加游戏难度选择和
排行榜界面
使用**多线程**实现音效的开启/关闭、
及火力道具

06

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	2	4	4
实验内容	飞机大战 功能分析	单例模式 工厂模式	Junit 单元测试	策略模式 数据访问对 象模式	Swing 多线程	观察者模式 模板模式
分数	4	6	4	6	6	14 (6+8)
提交内容	UML类图、 代码	UML类图、 代码	测试报告、 代码	UML类图、 代码	代码	项目代码、实 验报告、展示 视频

实验课程共16个学时，6个实验项目，总成绩为40分。

CONTENTS

目录

01 实验目的

02 实验任务

03 实验步骤

04 作业提交

实验目的

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



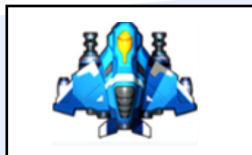
实验任务

1. 理解飞机大战系统功能，导入并运行模板程序；
2. 分析并设计普通和精英敌机类、道具类及子弹类，并使用PlantUML插件绘制相应的UML类图及继承关系；
3. 根据UML类图，重构模板代码，在项目中添加精英敌机类、三种道具类，以及它们的父类，完成本次迭代开发的目标。

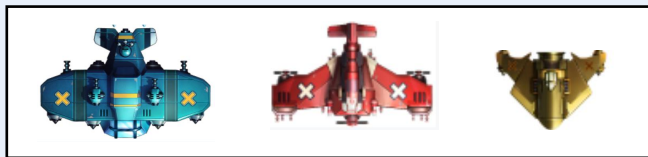
实验步骤：系统功能分析（1）

角色设定

- 英雄机



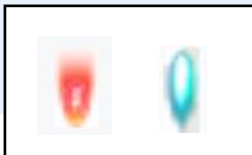
- 敌机



- 道具



- 子弹



实验步骤：系统功能分析（2）

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
- 生命值和得分计算、总分排行榜

实验步骤：系统功能分析（3）

界面和
音效

- 游戏地图
- 音效的开启和关闭

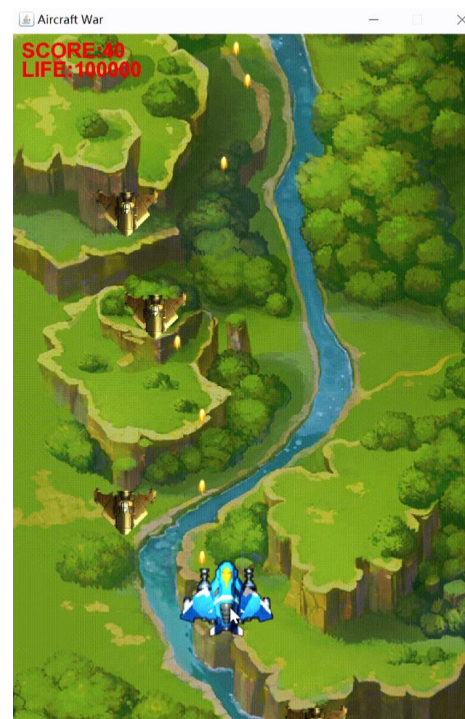
需求细则请参考实验指导书！

实验步骤：模板程序导入（1）

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

- 游戏主界面
- 英雄机、普通敌机移动
- 英雄机子弹直射
- 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。

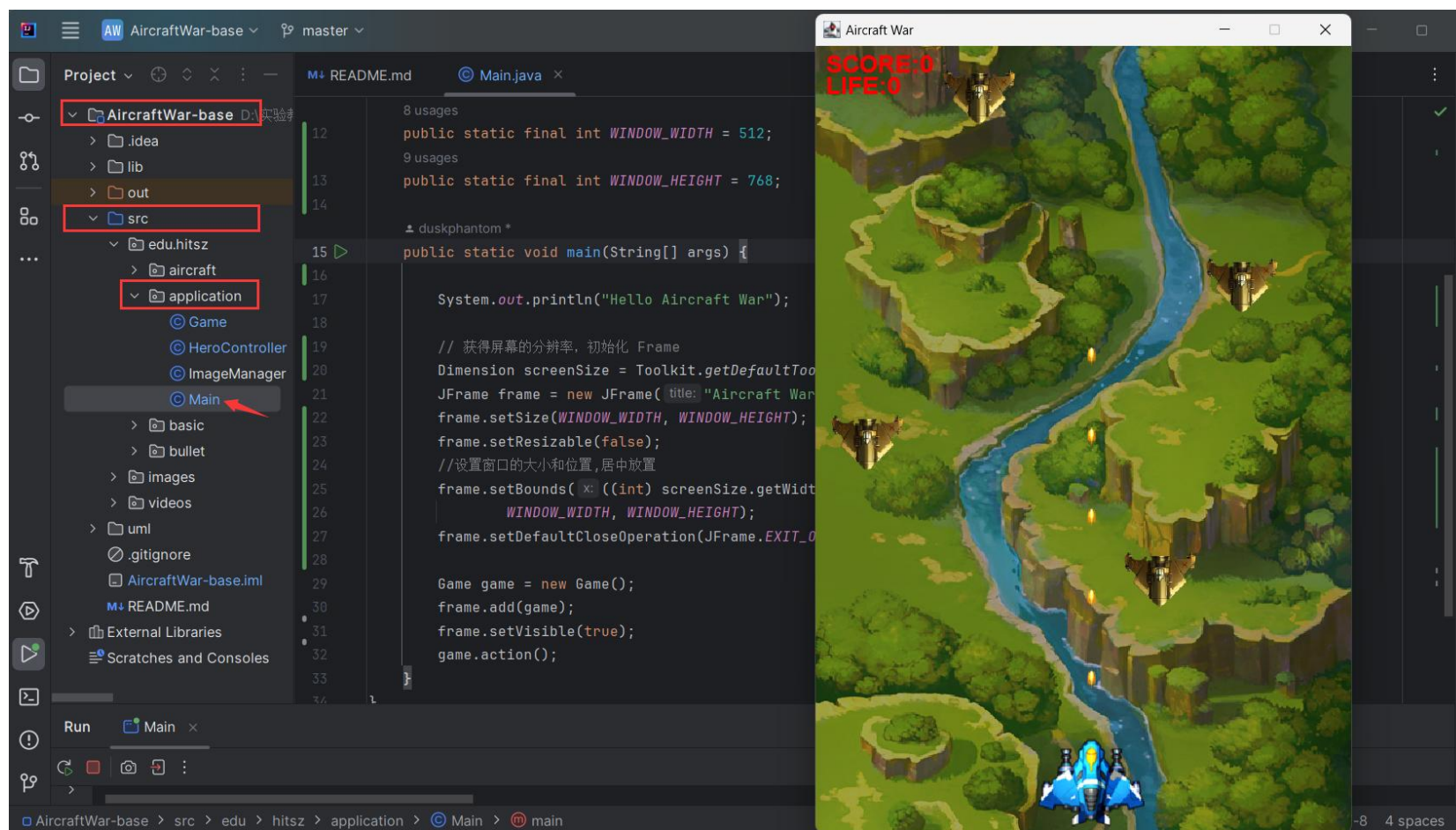


实验步骤：模板程序导入（2）

➤ 下载并安装 **IntelliJ IDEA Community** (社区版 2023.3.4)

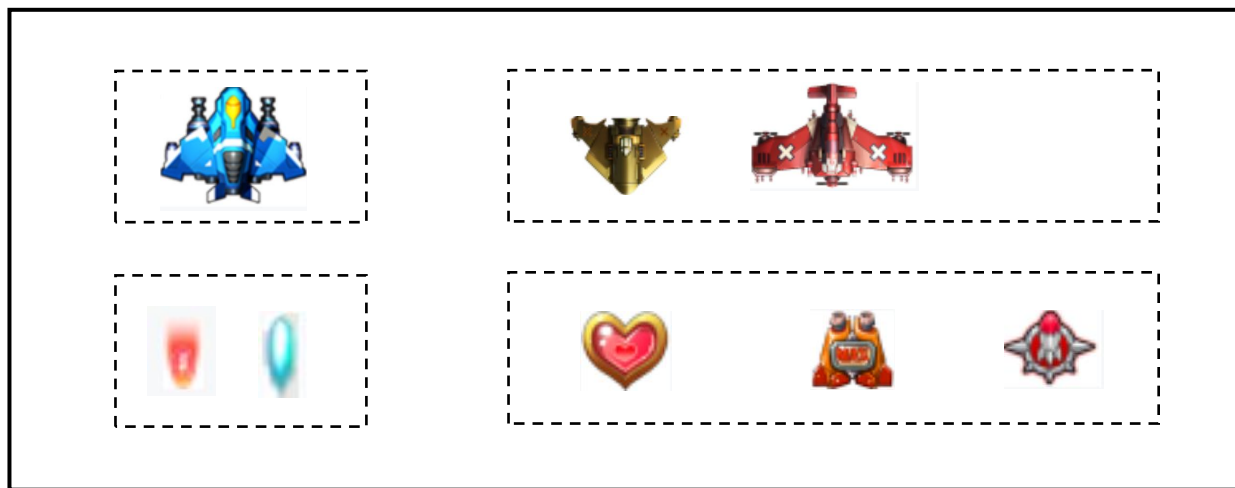
<https://www.jetbrains.com/zh-cn/idea/download/#section=windows>

➤ 运行 **Main.java**



实验步骤：UML类图（1）

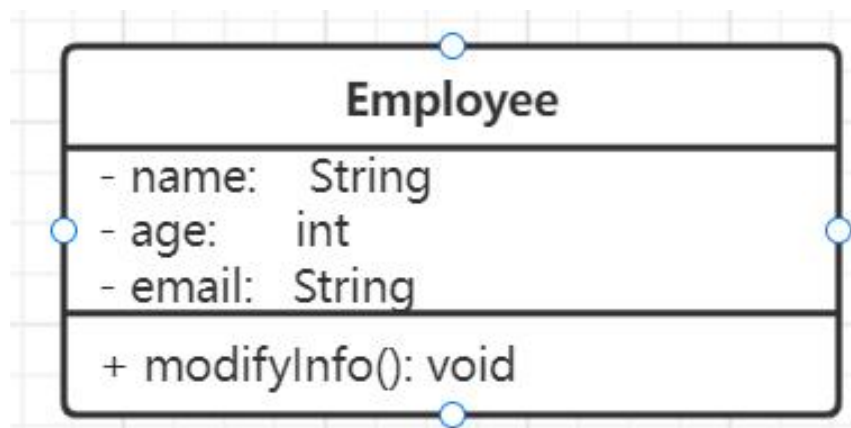
请根据面向对象设计原则，分析和设计游戏中的普通和精英敌机类、道具类和子弹类，并使用PlantUML插件绘制相应的UML类图及继承关系，类图中需包含英雄机、普通和精英敌机、三种道具、两种子弹及它们所继承的父类。



类图中可不包含Boss敌机

实验步骤：UML类图 (2)

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：UML类图（3）

类与类之间的关系： 关联、泛化、实现和依赖。

其中**关联**又分为**一般关联**、**聚合关系**和**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：老师和学生。

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。

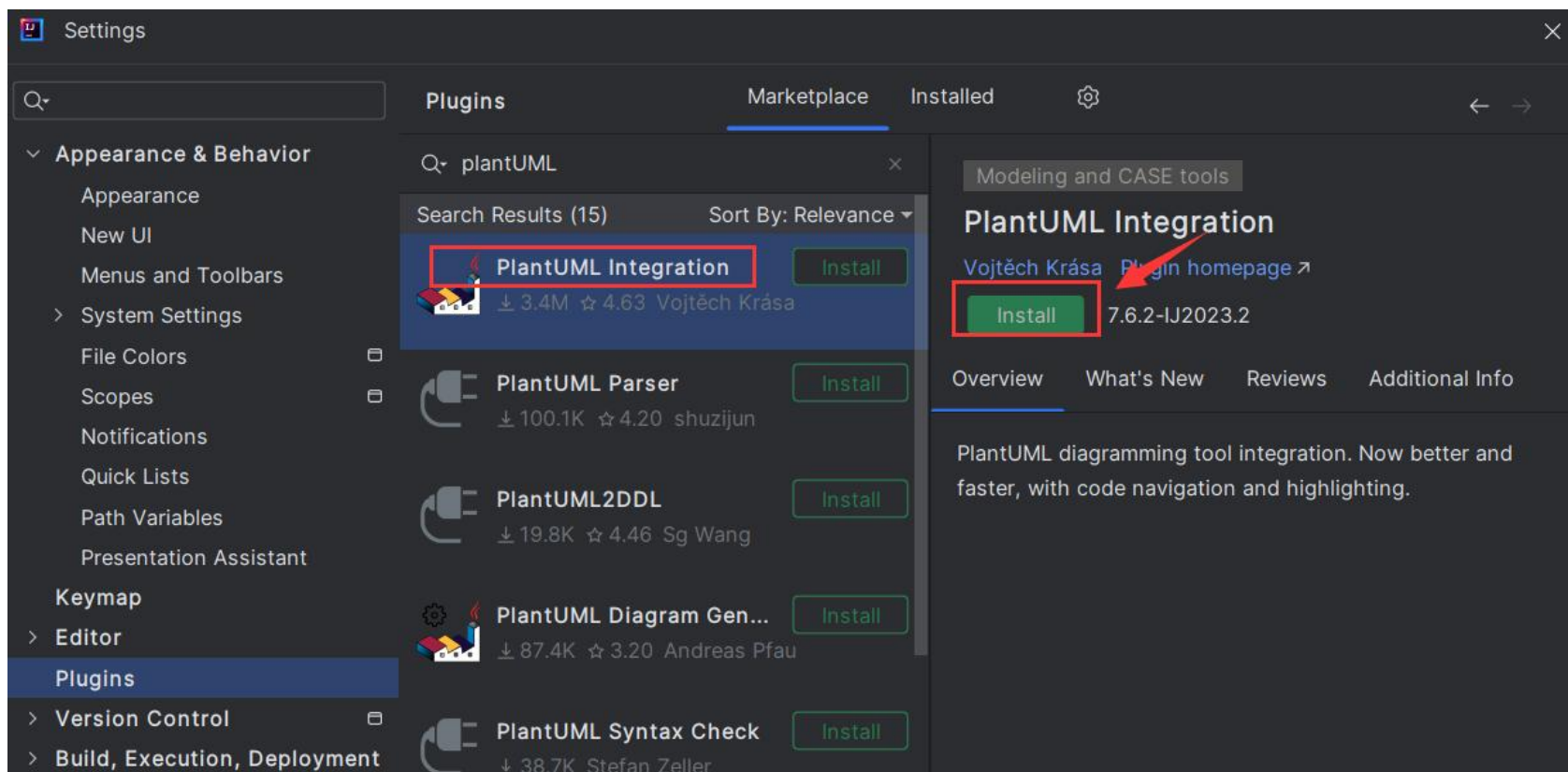
06 ↓

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：PlantUML插件（1）

- **PlantUML**是一个开源项目，支持快速绘制类图、时序图、用例图等UML结构图和行为图。



实验步骤：PlantUML插件（2）

➤ **PlantUML** 类图及相关语法请参考官网说明：

<https://plantuml.com/class-diagram>

➤ 接口、抽象类、类的画法：

```
@startuml
'https://plantuml.com/class-diagram

interface 接口
{
}

abstract class 抽象类
{
}

class 类
{
    + 公有成员变量: String
    - 私有成员变量: int
    # 受保护的成员变量: Date
    + {static} 静态成员变量: String

    + 构造函数(int 参数1)
    + 普通方法(int 参数1, String 参数2): void
    + {abstract} 抽象方法(): int
    + {static} 静态方法(int 参数1, String 参数2): String
}
```



接口



抽象类

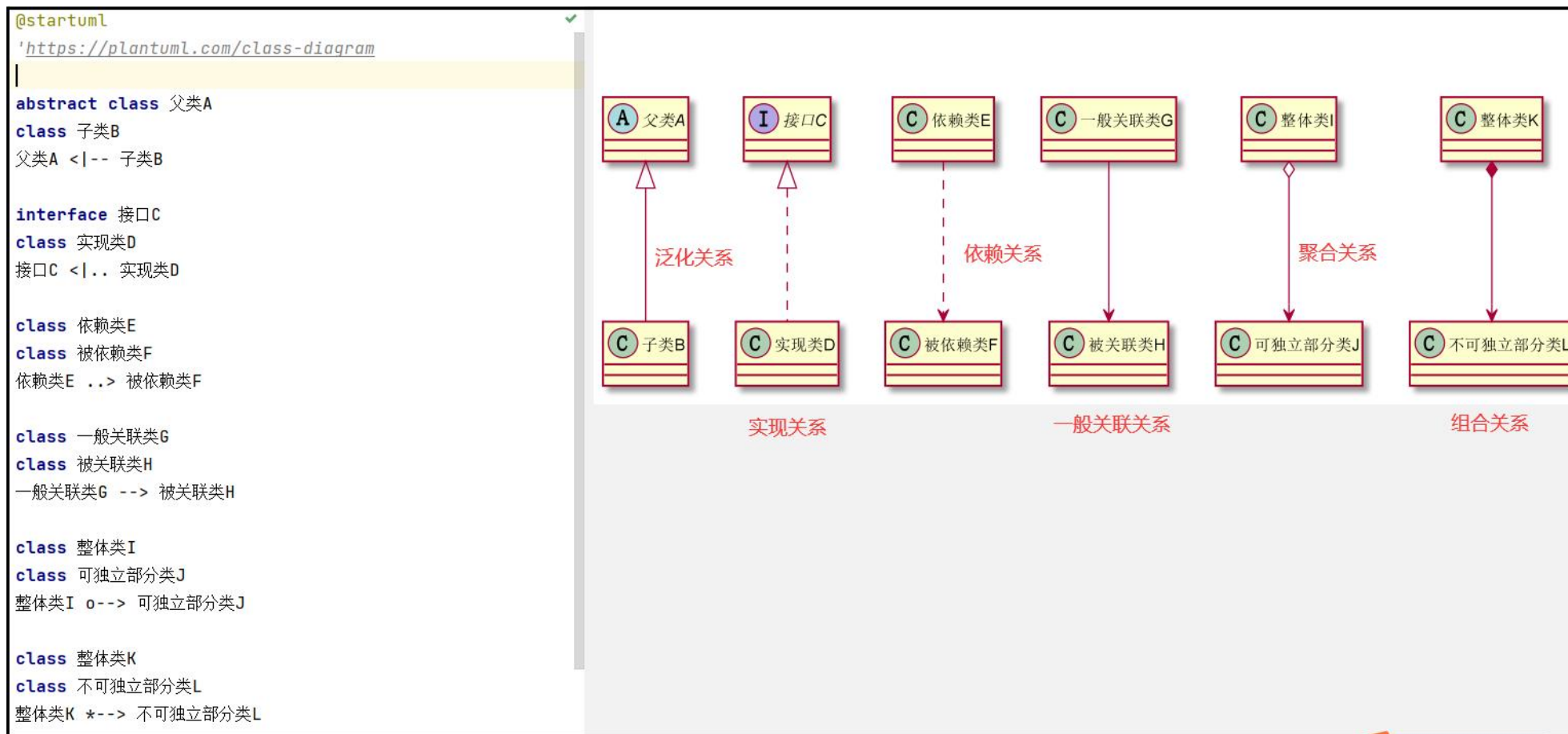


类

- 公有成员变量: String
- 私有成员变量: int
- ◇ 受保护的成员变量: Date
- 静态成员变量: String
- 构造函数(int 参数1)
- 普通方法(int 参数1, String 参数2): void
- 抽象方法(): int
- 静态方法(int 参数1, String 参数2): String

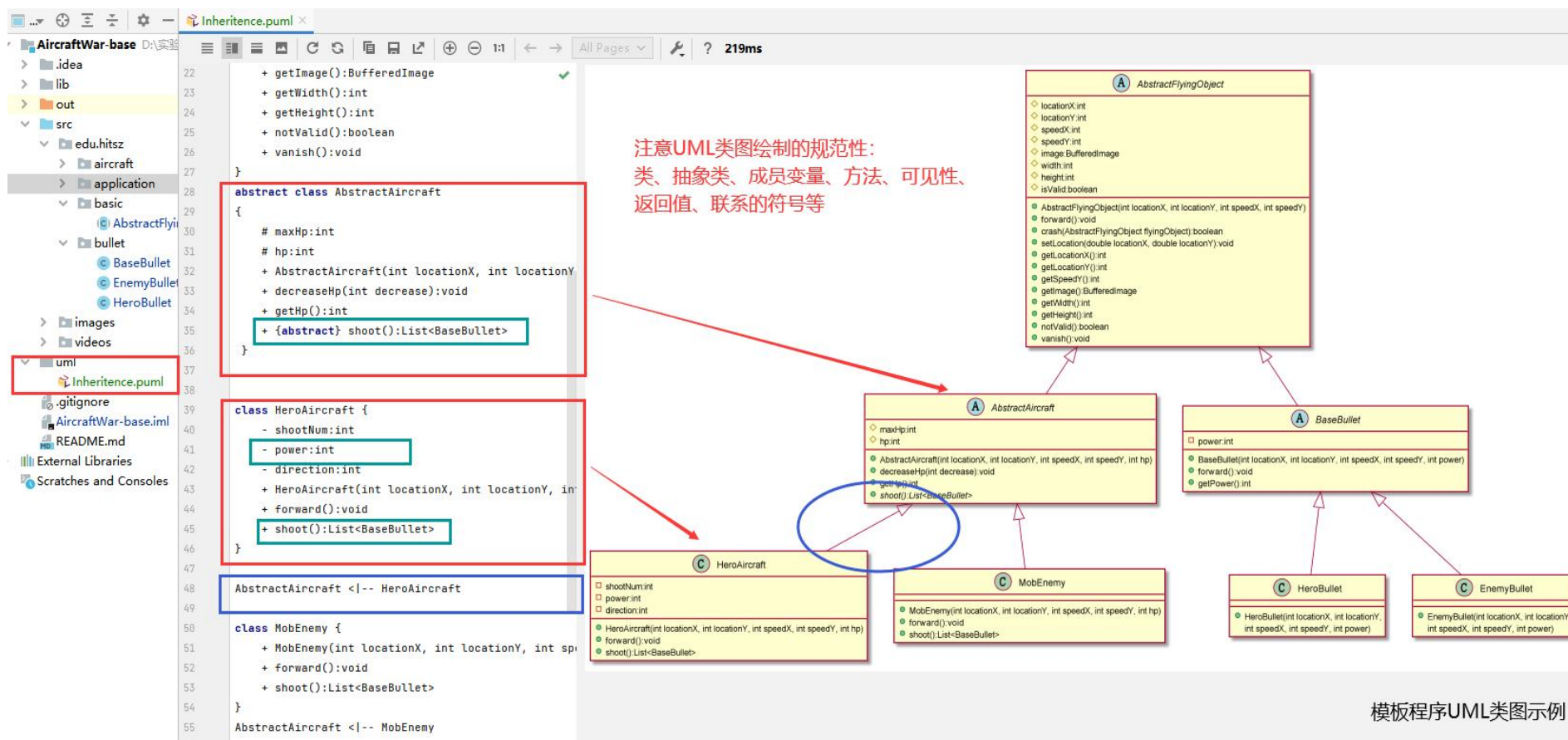
实验步骤：PlantUML插件 (3)

➤ 类与类之间6种关系的画法：



实验步骤：PlantUML插件（4）

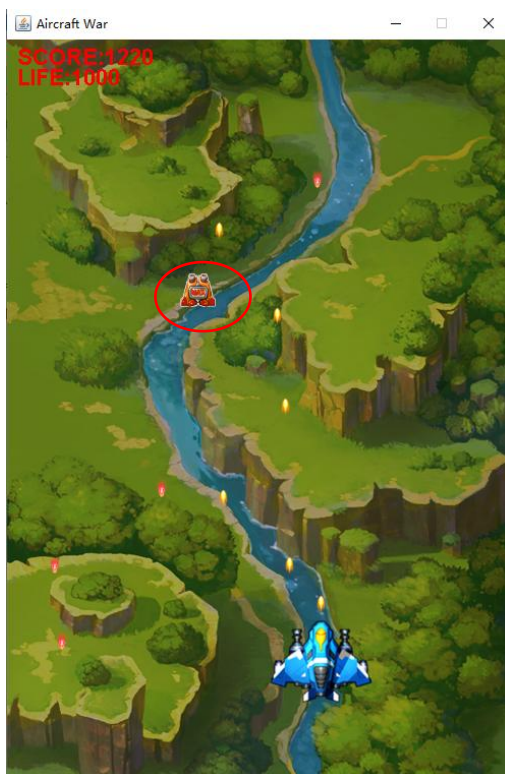
➤ 注意：请严格按照指导书4.3.1和4.3.2节要求绘制UML类图。



可通过在箭头内部使用关键字left, right, up 或 down来改变方向。（如：foo -left-> dummyLeft）

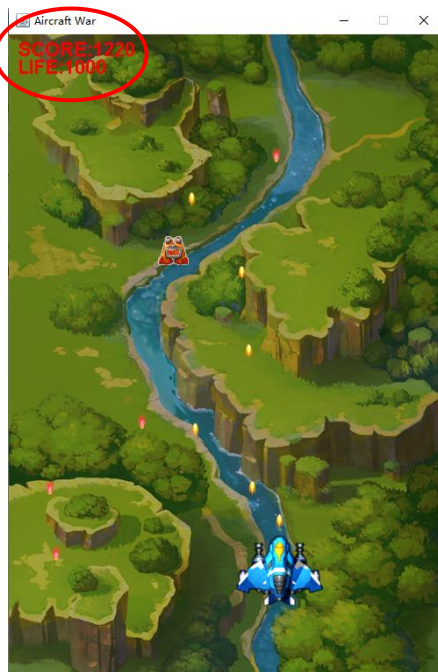
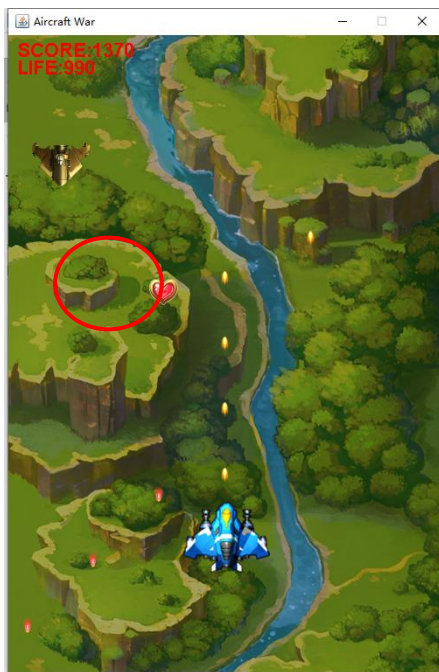
本次迭代开发的目标 (1)

- ✓ 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**直射**子弹;
- ✓ **精英**敌机坠毁后**随机**产生某种向下飞行的**道具** (或不产生) ;



本次迭代开发的目标 (2)

- ✓ 英雄机碰撞道具后，**道具**自动**生效**；
- ✓ **加血**道具可使英雄机恢复一定血量，但不超过初始值；
- ✓ **火力**道具和**炸弹**道具无需具体实现，生效时只需在控制台打印 “FireSupply active!” “BombSupply active!” 语句即可。



Run: Main

↑	89400
↓	90000
↕	90600
↖	91200
↗	91800
↘	92400
↙	93000
↕	93600
↖	94200
↗	BombSupply active
↘	94800

注意:

- ① 本实验暂不实现Boss机，UML类图和代码中无需包含Boss机类。
- ② 只需完成本次实验的基本要求，其他功能将在后续实验中逐步完善。

作业提交

- 提交内容

整个项目压缩包（压缩成zip包），包含：

- 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>

注意：上传后请自行下载确认是否提交成功。



同学们，请开始实验吧！

THANK YOU