

哈尔滨工业大学（深圳）

面向对象的软件构造导论 实验指导书

实验一 飞机大战功能分析

2022 春

目录

- 1. 实验目的.....3
- 2. 实验环境.....3
- 3. 实验内容（2 学时）3
- 4. 实验步骤.....3
 - 4.1 分析飞机大战系统功能.....3
 - 4.2 导入飞机大战模板程序.....6
 - 4.3 使用 PlantUML 插件设计敌机和道具类.....11
 - 4.3.1 UML 类图11
 - 4.3.2 类与类之间的关系.....12
 - 4.3.3 PlantUML 插件14
 - 4.4 在模板程序基础上，增加其他敌机类和道具类.....18
- 5. 实验提交.....18

1. 实验目的

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

2. 实验环境

- 1. Windows 10
- 2. IntelliJ IDEA 2021.2.3
- 3. Java 11

3. 实验内容（2 学时）

- 1. 分析飞机大战系统功能；
- 2. 导入飞机大战模板程序；
- 3. 根据面向对象设计原则，设计敌机和道具类，并使用 PlantUML 插件绘制其类图及继承关系；
- 4. 在模板程序的基础上增加其他敌机和道具类。

4. 实验步骤

4.1 分析飞机大战系统功能

根据下发的飞机大战任务书，分析系统中的类和对象，初步确定系统的功能模块。（任务书是本学期实验的总任务，不仅是本次实验的任务）

飞机大战任务书

1. 角色设定

| 名称：英雄机 | | 预期应用设计模式 |
|--------|---------|----------|
| 实例数量 | 1 | 单例模式 |
| 种类 | 1 种：英雄机 | |

| | | |
|------|--------------------------------------|------|
| 移动方式 | 跟随玩家鼠标移动 | |
| 生存 | 通过生命值（血）生存，被敌机子弹击中损失部分生命值，被敌机碰撞则全部损失 | |
| 火力 | 自动发射子弹，如有火力道具加成则改变弹道 | 策略模式 |
| 消灭 | 生命值为 0 时判定游戏失败 | |

| 名称：敌机 | | 预期应用设计模式 |
|-------|--|----------|
| 实例数量 | 不限 | 工厂模式 |
| 种类 | 3 种：普通敌机、精英敌机、Boss 敌机 | |
| 移动方式 | 普通敌机、精英敌机：以一定频率在界面随机位置出现并向屏幕下方移动 | |
| | Boss 敌机悬浮于界面上方，直至被消灭 | |
| 生存 | 通过生命值（血）生存，被英雄机子弹击中损失部分生命值，生命值为 0 时坠毁。 | |
| 火力 | 普通敌机不发射子弹，精英敌机和 Boss 机按照一定轨迹、频率模式发射子弹 | 策略模式 |
| 消灭 | 敌机坠毁时，总分增加相应分数，随机掉落某种道具 | |

| 名称：道具 | | 预期应用设计模式 |
|-------|---------------------|----------|
| 实例数量 | 不限 | 工厂模式 |
| 种类 | 3 种：火力道具、炸弹道具、加血道具 | |
| 出现方式 | 敌机坠毁后，以较低概率随机出现某种道具 | |
| 使用方式 | 英雄机碰撞后自动触发 | |

| | | |
|------|-------------------------------|-------|
| 移动方式 | 以一定速度向屏幕下方移动 | |
| 效果 | 火力道具：一段时间内英雄机改变弹道 | |
| | 炸弹道具：清除界面上除 boss 机外的所有敌机和敌机子弹 | 观察者模式 |
| | 加血道具：英雄机增加生命值 | |
| 消灭 | 与英雄机碰撞或移动至界面底部则消失 | |

| 名称：子弹 | | 预期应用设计模式 |
|-------|---------------------|----------|
| 实例数量 | 不限 | |
| 种类 | 2 种：英雄机子弹、敌机子弹 | |
| 速度 | 子弹飞行速度 | |
| 角度 | 子弹初始飞行方向 | |
| 移动方式 | 以一定速度向某个方向匀速直线运动 | |
| 消灭 | 击中英雄机、敌机，或飞出界面外自动消亡 | |

| 名称：排行榜 | | 预期应用设计模式 |
|--------|--|----------|
| 排名方式 | 根据游戏难度分类，按照总分排名 | 数据访问对象模式 |
| 记录信息 | 游戏难度、名次、玩家名、得分、记录时间 | |
| 更新方式 | 每局游戏结束后询问玩家是否存储记录，如存储则插入新的游戏记录，也可删除历史记录。 | |

2. 游戏规则

- (1) 用户进入游戏界面后，可选择某种**游戏难度**：简单 / 普通 / 困难。其中，普通和困难模式在得分达到设定阈值后，出现 Boss 机。

- (2) 英雄机、精英机和 Boss 机采用**不同弹道**发射子弹，如 Boss 机可散射。
- (3) 敌机或英雄机被**子弹击中**后（子弹消失），生命值减少，生命值为 0 时**坠毁**。不同类型敌机坠毁后，英雄机增加相应的分数。
- (4) 英雄机**生命值为 0** 或与**任意敌机相撞**，英雄机坠毁，**游戏结束**。
- (5) 精英机和 Boss 机坠毁后会以较低概率**随机掉落某种道具**（也可不产生道具），英雄机碰撞道具后，道具自动生效。
- (6) 普通和困难模式，每一局**随着游戏时长增加，程序会逐渐增加游戏难度**，如增加敌机血量、加快敌机速度、增加精英机出现概率、缩短敌机产生周期等。
- (7) 游戏过程中记录**英雄机得分**，在界面左上角显示。**游戏结束后**显示玩家当局分数，显示当前游戏难度的历史**得分排行榜**。询问玩家是否保存本次记录，如保存，插入新的游戏记录，需要玩家输入相关信息（如玩家姓名）。玩家也可删除历史记录。

3. 界面和音效需求

- (1) **游戏界面**：风格清新，简单易操作。
- (2) **音效**：可打开或关闭
 - ✧ 游戏背景音乐
 - ✧ Boss 机出现
 - ✧ 敌机被子弹击中
 - ✧ 三种道具生效

4.2 导入飞机大战模板程序

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

- ✓ 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级生命值

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

已完成 IntelliJ IDEA 安装的可以跳过本节。

(1) 使用教育邮箱(xxx@hit.edu.cn) 申请 JETBRAINS 免费的个人许可证。
申请网址: <https://www.jetbrains.com/zh-cn/community/education/#students>



面向学生和教师的个人许可证

获取所有 JetBrains IDE 的免费访问权限, 在学校或家中用于个人目的。

谁可以获得用于教育的免费个人许可证?

欢迎来自正规教育机构(高中、学院和大学)的学生和教职员工申请。

学生必须参加正规教育计划, 该计划需要一年或多年全日制学习才能完成计划。

我们需要您提供最近 6 个月的注册信息或成绩单。该文件必须包含您的姓名、学生证号、日期等信息。您还可以使用[学信网](#)的学校状态验证报告。

不确定许可证条款? 请查看[常见问题解答](#)或[在此处](#)阅读完整条款。

立即申请



JetBrains Products for Learning

Before you apply, please read the [Educational Subscription Terms and FAQ](#).

Apply with: UNIVERSITY EMAIL ADDRESS ISIC/ITIC MEMBERSHIP OFFICIAL DOCUMENT GITHUB

Status: ☒ I'm a student ☐ I'm a teacher

Level of study: Undergraduate

Is Computer Science or Engineering your major field of study? ☒ Yes ☐ No

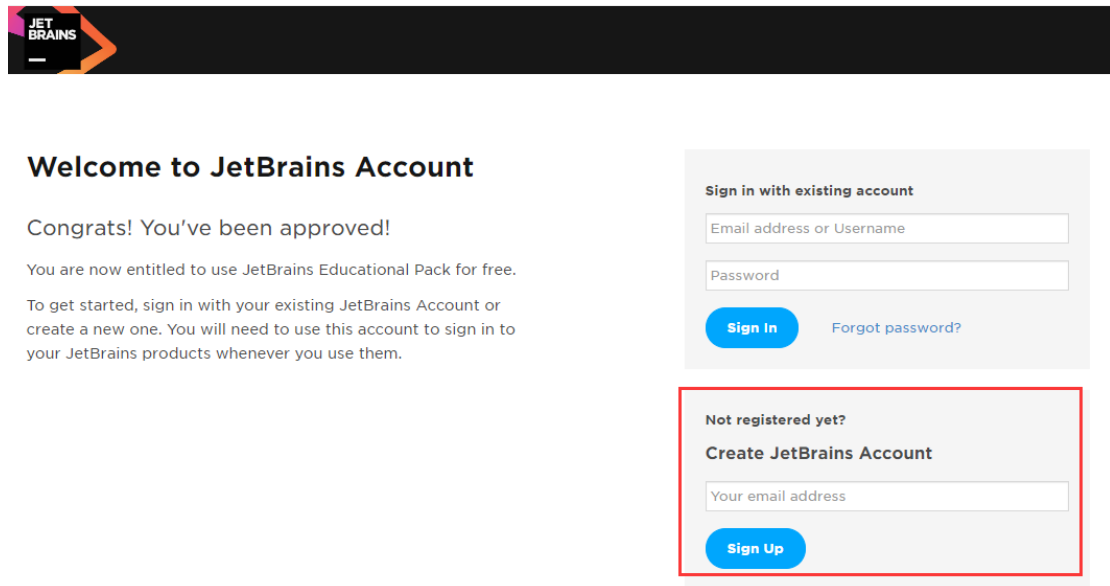
Graduation date: Choose

Choose expected graduation date

Email address: University email address, e.g. js@mit.edu

I certify that the university email address provided above is valid and belongs to me.

申请后在邮箱中点击接受许可证协议，即可注册 JetBrains Account。



Welcome to JetBrains Account

Congrats! You've been approved!

You are now entitled to use JetBrains Educational Pack for free.

To get started, sign in with your existing JetBrains Account or create a new one. You will need to use this account to sign in to your JetBrains products whenever you use them.

Sign in with existing account

Email address or Username

Password

Sign In [Forgot password?](#)

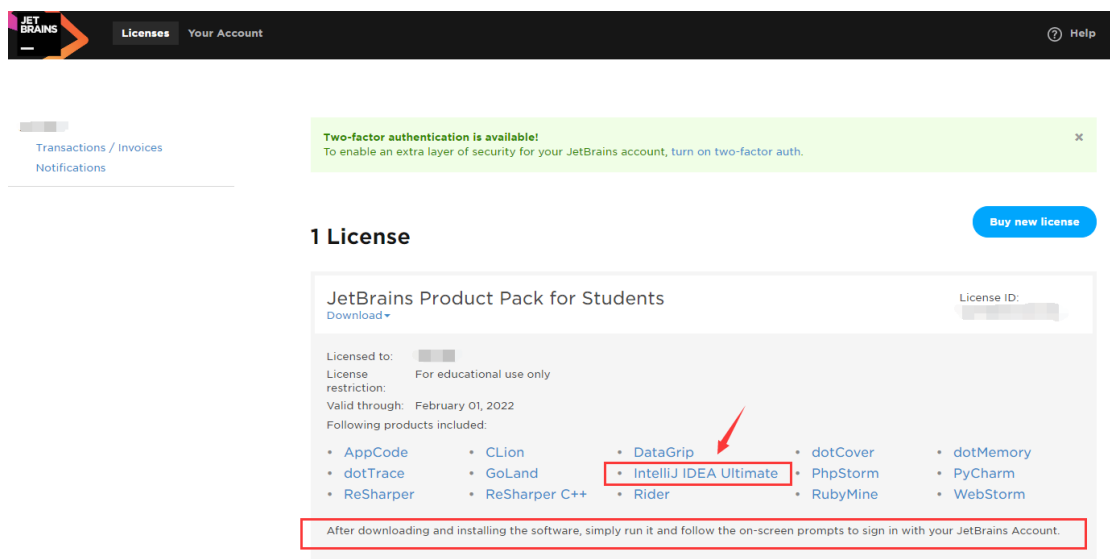
Not registered yet?

Create JetBrains Account

Your email address

Sign Up

(2) 下载 IntelliJ IDEA Ultimate 安装包，并根据系统配置选择对应的版本安装。



Licenses Your Account [Help](#)

Transactions / Invoices
Notifications

Two-factor authentication is available!
To enable an extra layer of security for your JetBrains account, turn on two-factor auth.

1 License [Buy new license](#)

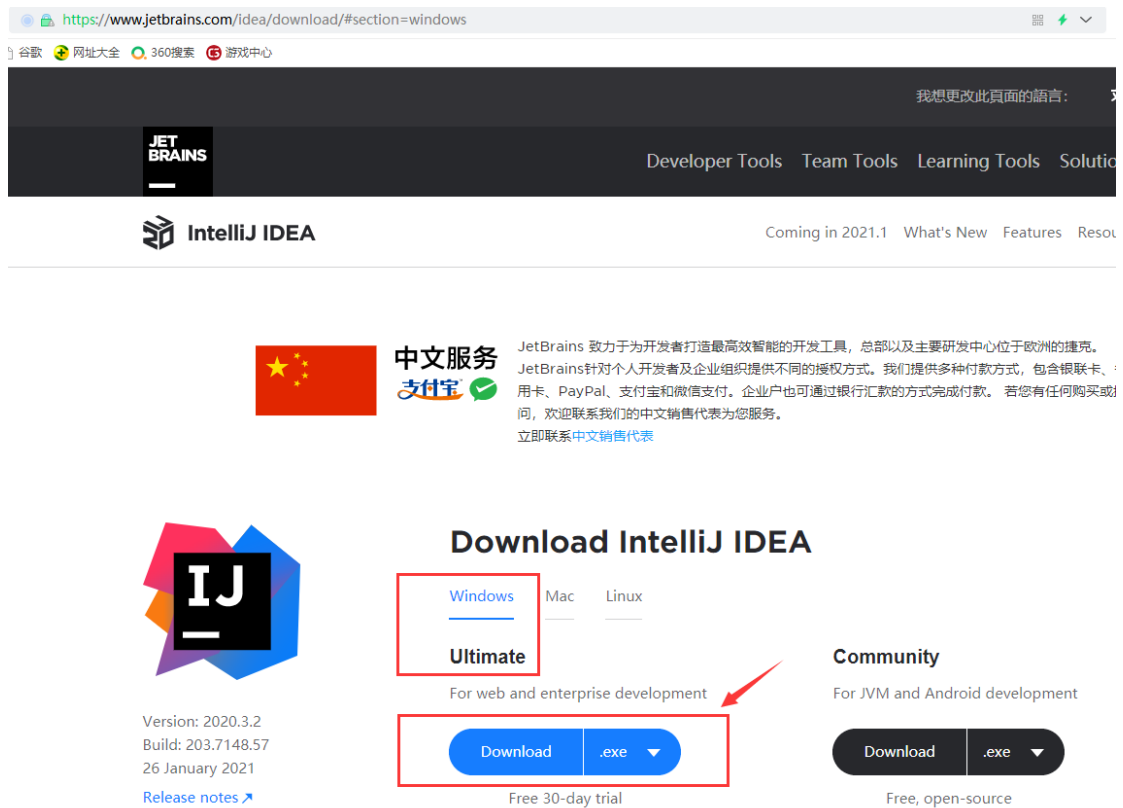
JetBrains Product Pack for Students License ID: [redacted]
[Download](#)

Licensed to: [redacted]
License restriction: For educational use only
Valid through: February 01, 2022
Following products included:

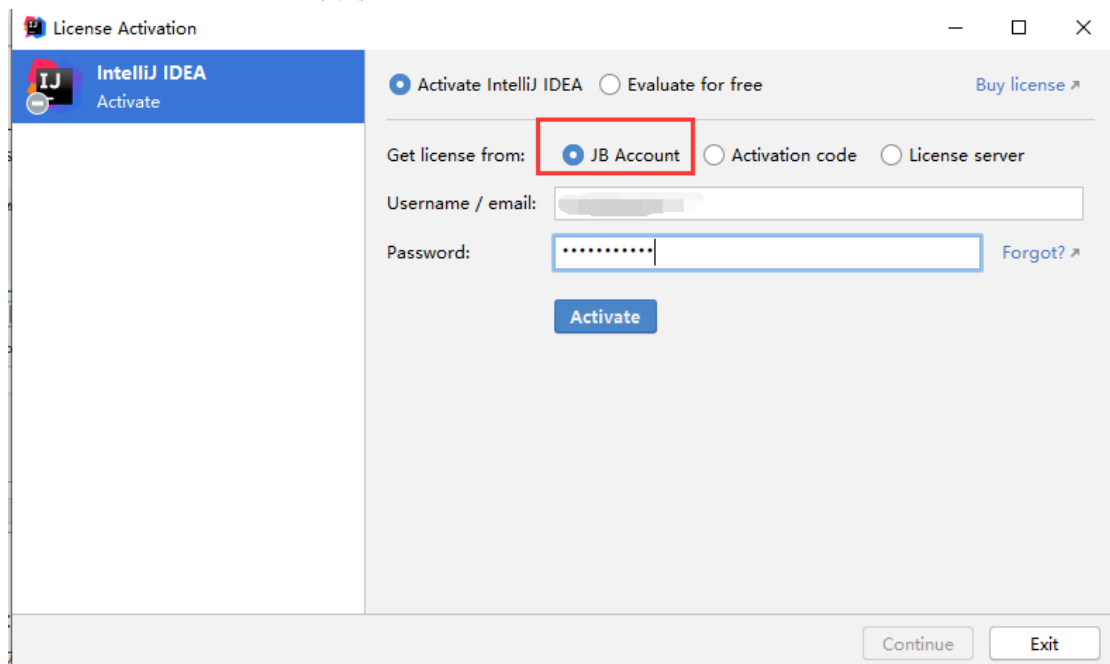
- AppCode
- CLion
- DataGrip
- dotCover
- dotMemory
- dotTrace
- GoLand
- **IntelliJ IDEA Ultimate**
- PhpStorm
- PyCharm
- ReSharper
- ReSharper C++
- Rider
- RubyMine
- WebStorm

After downloading and installing the software, simply run it and follow the on-screen prompts to sign in with your JetBrains Account.












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

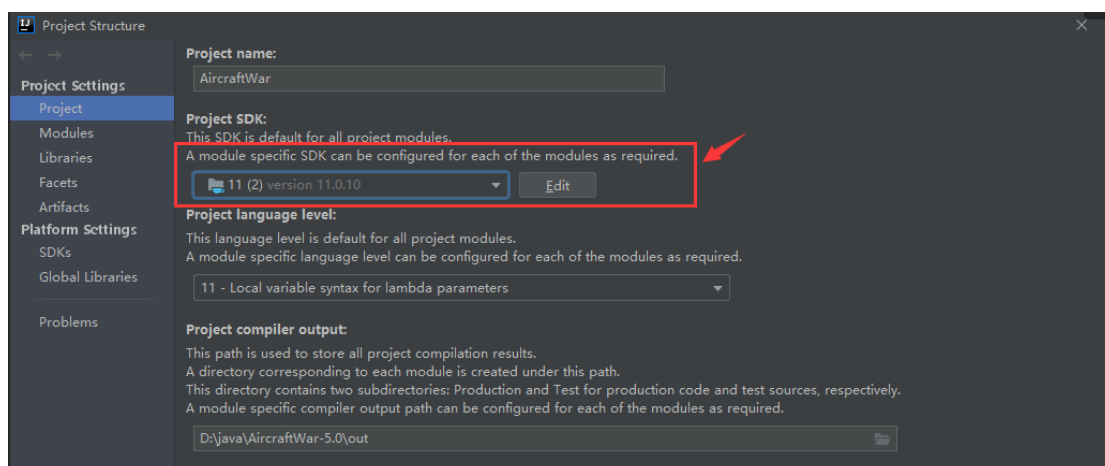


(3) 运行 IntelliJ IDEA 并使用注册的 JetBrains Account 激活。

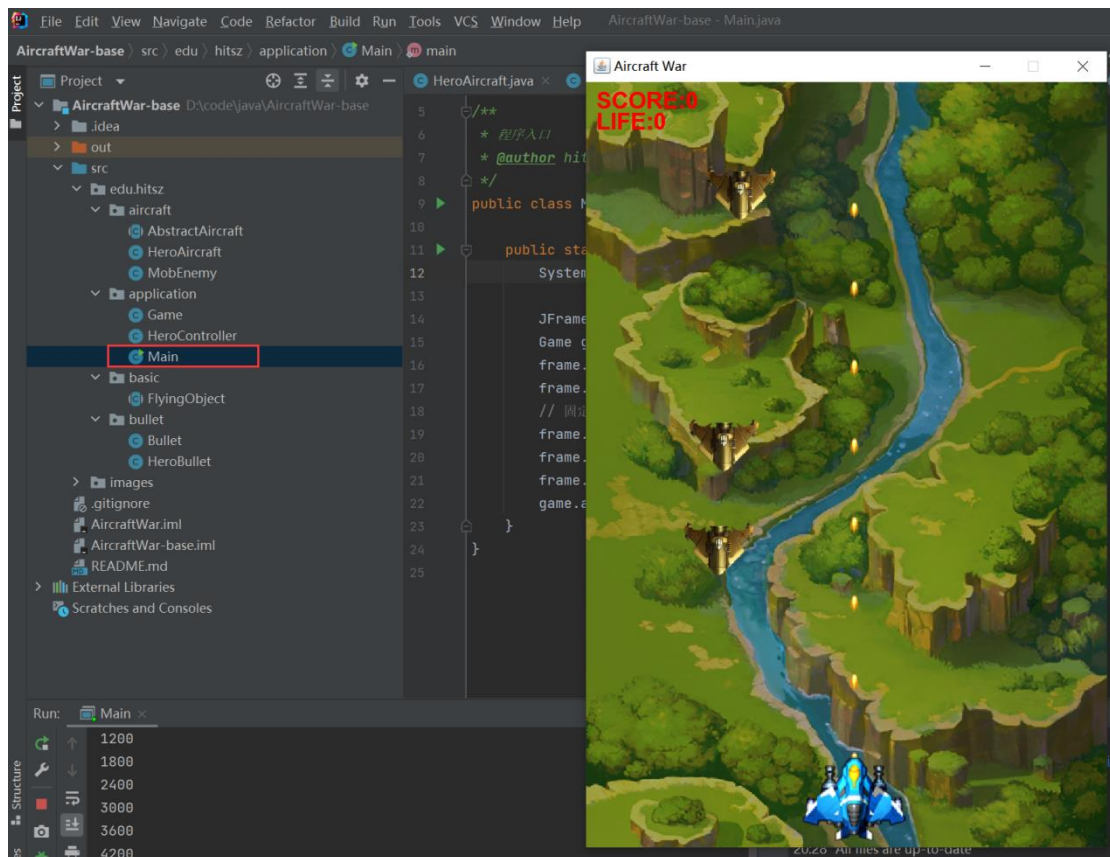


(4) 下载并安装 JDK11，新建或打开 Project 时，选择 Project SDK 为 JDK11。
下载地址: <https://www.oracle.com/java/technologies/javase-jdk11-downloads.html>

| Java SE Development Kit 11.0.10 | | |
|--|-----------|---|
| This software is licensed under the Oracle Technology Network License Agreement for Oracle Java SE | | |
| Product / File Description | File Size | Download |
| Linux ARM 64 Debian Package | 145.64 MB |  jdk-11.0.10_linux-aarch64_bin.deb |
| Linux ARM 64 RPM Package | 152.22 MB |  jdk-11.0.10_linux-aarch64_bin.rpm |
| Linux ARM 64 Compressed Archive | 169.37 MB |  jdk-11.0.10_linux-aarch64_bin.tar.gz |
| Linux x64 Debian Package | 149.39 MB |  jdk-11.0.10_linux-x64_bin.deb |
| Linux x64 RPM Package | 156.12 MB |  jdk-11.0.10_linux-x64_bin.rpm |
| Linux x64 Compressed Archive | 173.31 MB |  jdk-11.0.10_linux-x64_bin.tar.gz |
| macOS Installer | 167.51 MB |  jdk-11.0.10_osx-x64_bin.dmg |
| macOS Compressed Archive | 167.84 MB |  jdk-11.0.10_osx-x64_bin.tar.gz |
| Solaris SPARC Compressed Archive | 184.82 MB |  jdk-11.0.10_solaris-sparcv9_bin.tar.gz |
| Windows x64 Installer | 152.32 MB |  jdk-11.0.10_windows-x64_bin.exe |
| Windows x64 Compressed Archive | 171.67 MB |  jdk-11.0.10_windows-x64_bin.zip |



(5) 导入项目 AircraftWar-base 后运行 `src\edu\hitsz\application\Main.java`，模板代码已完成基本运行框架，并已创建英雄机和普通敌机。

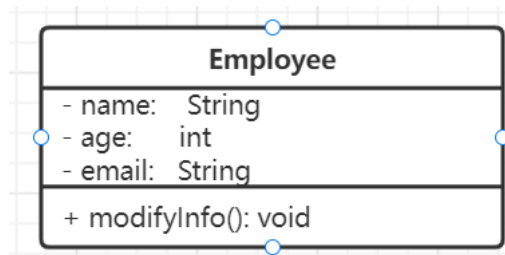


4.3 使用 PlantUML 插件设计敌机和道具类

游戏中有 3 种类型的敌机：普通敌机、精英敌机、Boss 敌机，3 种类型的道具：火力道具、炸弹道具、加血道具。请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、所有敌机和道具类及它们所继承的父类。

4.3.1 UML 类图

类图（Class Diagram）：类图是面向对象系统建模中最常用和最重要的图，是定义其它图的基础。类图主要是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。在 UML 中，类使用包含类名、类的属性和操作且带有分隔线的长方形来表示，如定义一个 Employee 类，它包含属性 name、age 和 email，以及操作 modifyInfo()。



对应的 Java 代码片段如下：

```

public class Employee {
    private String name;
    private int age;
    private String email;

    public void modifyInfo() {
        .....
    }
}
  
```

4.3.2 类与类之间的关系

在画类图的时候，理清类和类之间的关系是重点。类的关系有关联 (Association)、泛化 (Generalization)、实现 (Realization) 和依赖 (Dependency)。其中关联又分为一般关联关系和聚合关系 (Aggregation)，组合关系 (Composition)。

● 关联 (Association)

它是类与类之间最常用的一种关系，它是一种结构化关系，用于表示一类对象与另一类对象之间有联系。

① 一般关联

【含义】：是一种拥有的关系，它使一个类知道另一个类的属性和方法；可以是双向的，也可以是单向的。如老师和学生、顾客和商品等。

【代码体现】：成员变量

【箭头及指向】：带普通箭头的实心线，指向被拥有者。双向的关联可以有

两个箭头或者没有箭头，单向的关联有一个箭头。

② 聚合 (Aggregation) 关系

【含义】：是整体与部分的关系，且部分可以离开整体而单独存在。如汽车和轮胎是整体和部分的关系，轮胎离开车仍然可以存在。聚合关系是关联关系的一种，是强的关联关系；关联和聚合在语法上无法区分，必须考察具体的逻辑关系。

【代码体现】：成员变量

【箭头及指向】：带空心菱形的实心线，菱形指向整体。



③ 组合（Composition）关系

【含义】：是整体与部分的关系，但部分不能离开整体而单独存在。如公司和部门是整体和部分的关系，没有公司就不存在部门。组合关系是关联关系的一种，是比聚合关系还要强的关系，它要求普通的聚合关系中代表整体的对象负责代表部分的对象的生命周期。

【代码体现】：成员变量

【箭头及指向】：带实心菱形的实线，菱形指向整体



● 泛化（Generalization）

【含义】：也就是继承关系，用于描述父类与子类之间的关系，父类又称基类或超类，子类又称作派生类。表示一般与特殊的关系，它指定了子类如何特化父类的所有特征和行为。例如：老虎是动物的一种，即有老虎的特性也有动物的共性。

【箭头指向】：带三角箭头的实线，箭头指向父类



● 实现（Realization）

【含义】：是一种类与接口的关系，表示类是接口所有特征和行为的实现者。例如：定义了一个交通工具接口 `Vehicle`，包含一个抽象操作 `move()`，在类 `Ship` 和类 `Car` 中都实现了该 `move()` 操作，不过具体的实现细节将会不一样。`Ship` 和 `Car` 类与接口 `Vehicle` 是实现关系。

【箭头指向】：带三角箭头的虚线，箭头指向接口



● 依赖（Dependency）

【含义】：是一种使用关系，特定事物的改变有可能会影响到使用该事物的其他事物，在需要表示一个事物使用另一个事物时使用依赖关系。例如：驾驶员开车，在 `Driver` 类的 `drive()` 方法中将 `Car` 类型的对象 `car` 作为一个参数传递，以便在 `drive()` 方法中能够调用 `car` 的 `move()` 方法，且驾驶员的 `drive()` 方法依赖车的 `move()` 方法，因此类 `Driver` 依赖类 `Car`。

【代码表现】：局部变量、方法的参数或者对静态方法的调用

【箭头及指向】：带箭头的虚线，指向被使用者



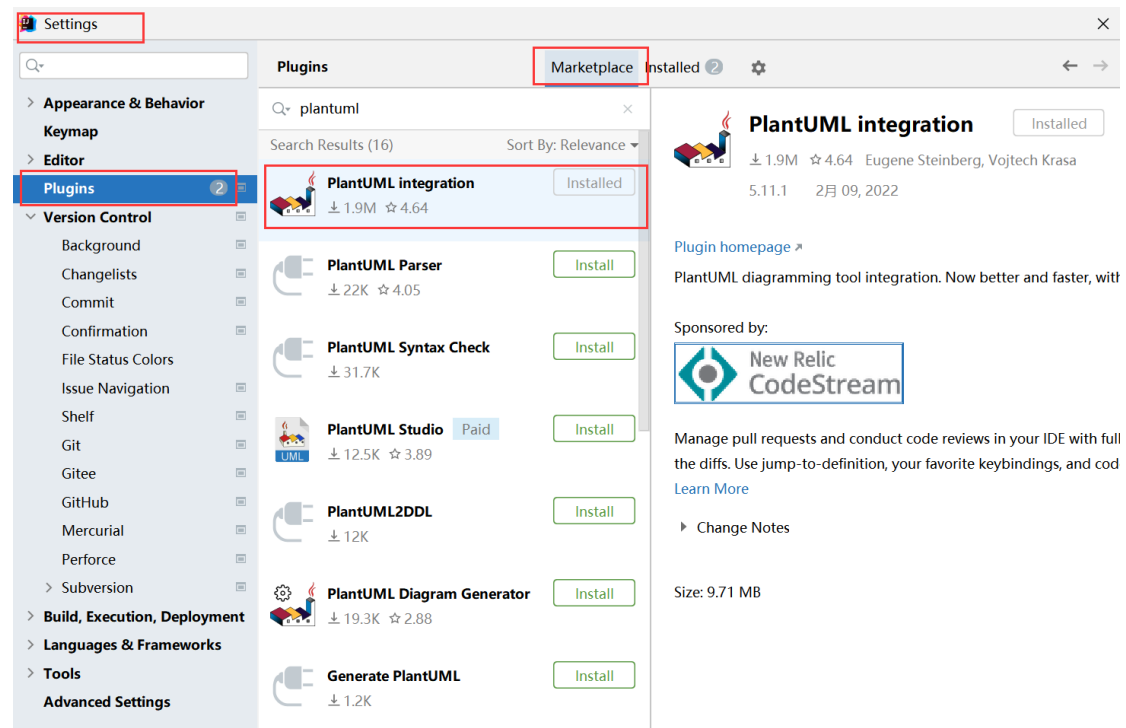
4.3.3 PlantUML 插件

PlantUML 是一个开源项目，支持快速绘制时序图、用例图、类图、对象图、活动图、组件图、部署图、状态图和定时图。Plantuml 本质上是也算一门可以快速画图的设计语言，通过简单的描述语言绘制，非常符合程序员的习惯。

(1) IntelliJ IDEA 安装插件

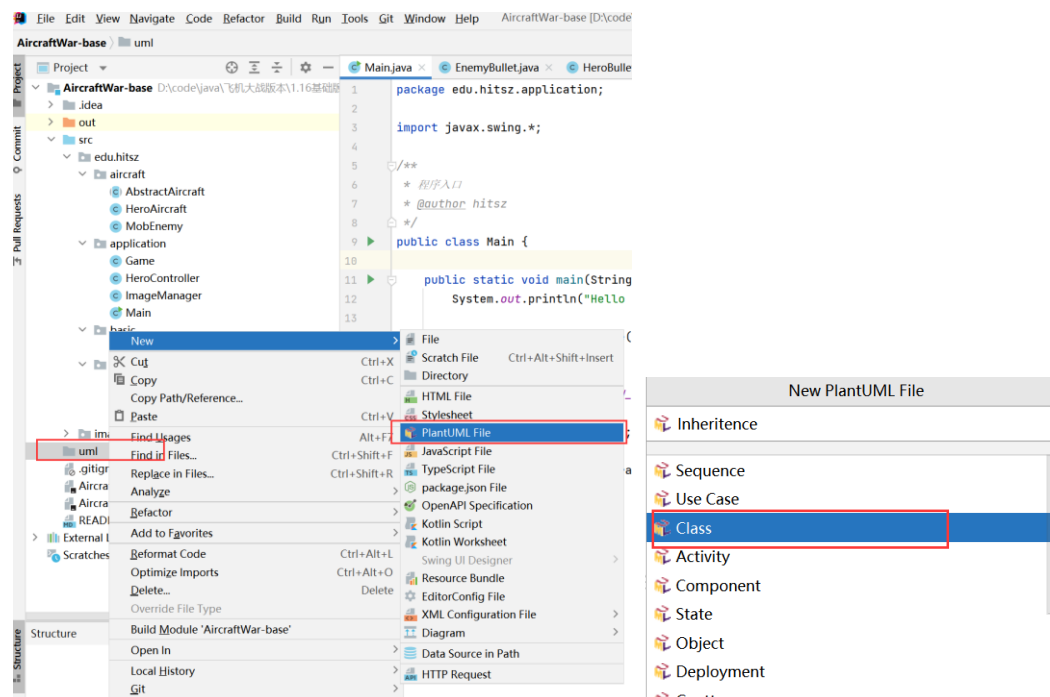
Plantuml 提供了 IntelliJ IDEA 插件，通过 IDEA 的插件管理搜索 PlantUml 并集成安装后，重启 IDEA 即可。

点击 File --> Settings --> Plugins --> Marketplace，搜索 PlantUml。



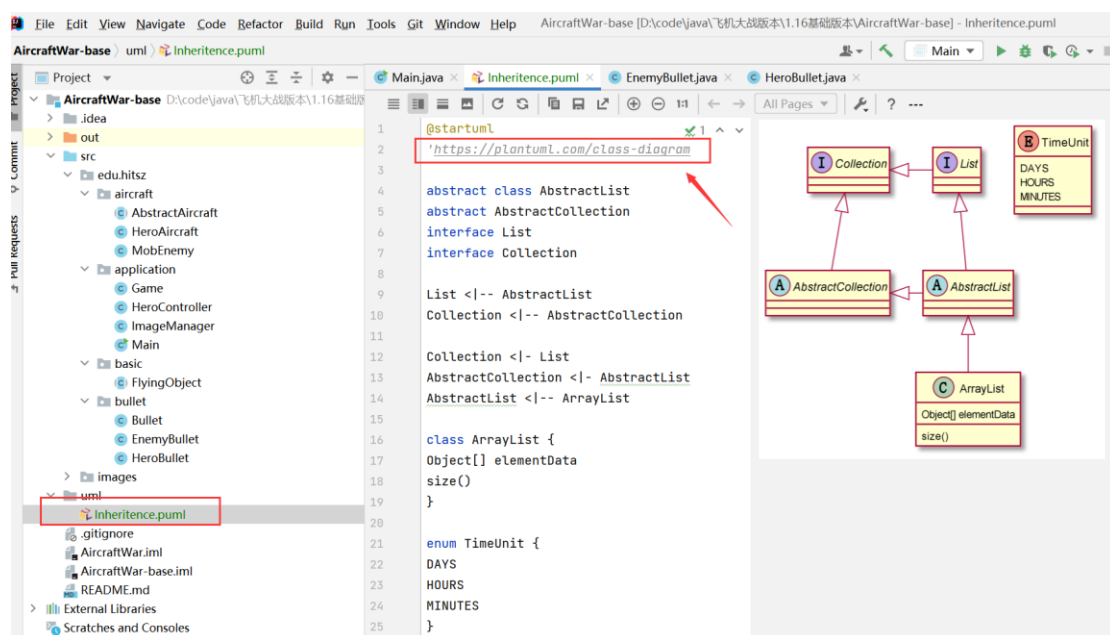
(2) PlantUML 绘制类图示例

在项目下新建 uml 文件夹，右键选择 New-->PlantUML File 创建文件，选择 Class 类图。puml 文件可以像代码一样在 IDEA 中进行管理，即通过 IDEA 可以完成日常设计，开发过程中的大多数工作。



(3) PlantUML 类图的语法和功能

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





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。

(4) 飞机大战模板程序类图示例

Inheritance.puml

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

abstract class FlyingObject
{
    # int locationX
    # int locationY
    # int speedX
    # int speedY
    # BufferedImage image
    # int width
    # int height
    # boolean isValid

    + FlyingObject(int locationX, int locationY, int speedX, int speedY)
    + void forward()
    + boolean crash(FlyingObject flyingObject)
    + void setLocation(double locationX, double locationY)
    + int getLocationX()
    + int getLocationY()
    + int getSpeedY()
    + BufferedImage getImage()
    + int getWidth()
    + int getHeight()
    + boolean notValid()
    + void vanish()
}

abstract class AbstractAircraft
{
    # int maxHp
    # int hp
    + AbstractAircraft(int locationX, int locationY, int speedX, int speedY, int hp)
    + void decreaseHp(int decrease)
    + int getHp()
    + {abstract} List<Bullet> shoot()
}

abstract class AbstractBullet
{
```



```

        - int power
        + Bullet(int locationX, int locationY, int speedX, int speedY, int power)
        + void forward()
        + int getPower()
    }

    FlyingObject <|-- AbstractAircraft
    FlyingObject <|-- AbstractBullet

    AbstractAircraft <|-- HeroAircraft
    AbstractAircraft <|-- MobEnemy

    AbstractBullet <|-- HeroBullet
    AbstractBullet <|-- EnemyBullet

    class HeroAircraft {
        - int shootNum
        - int power
        - int direction
        + HeroAircraft(int locationX, int locationY, int speedX, int speedY, int hp)
        + void forward()
        + List<Bullet> shoot()
    }

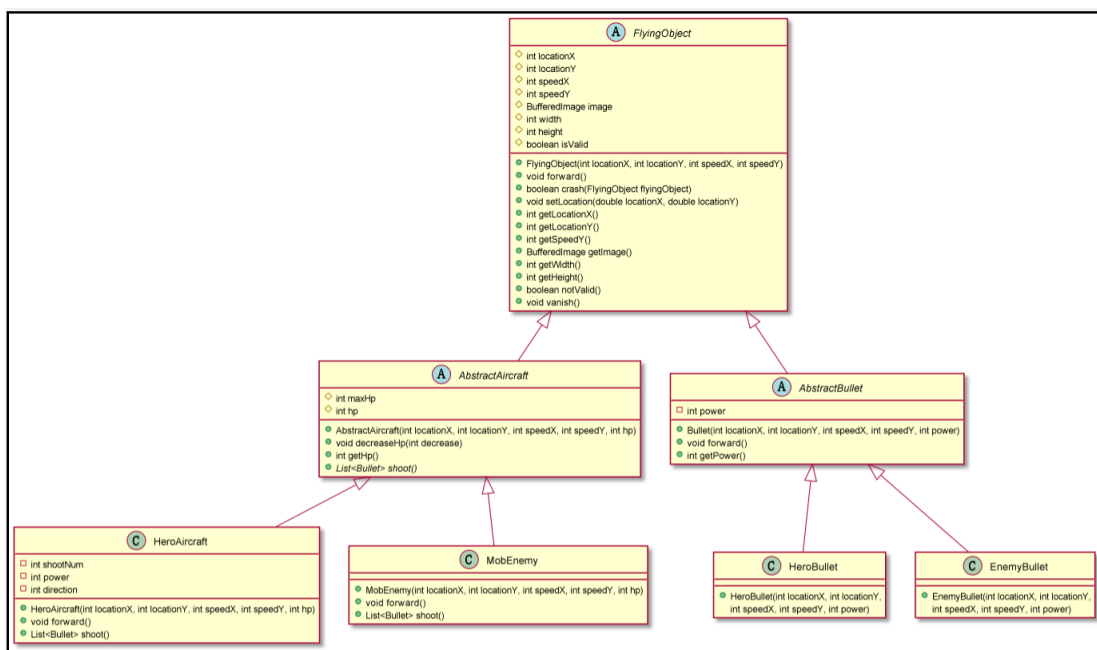
    class MobEnemy {
        + MobEnemy(int locationX, int locationY, int speedX, int speedY, int hp)
        + void forward()
        + List<Bullet> shoot()
    }

    class HeroBullet {
        + HeroBullet(int locationX, int locationY,
            int speedX, int speedY, int power)
    }

    class EnemyBullet {
        + EnemyBullet(int locationX, int locationY,
            int speedX, int speedY, int power)
    }

    @enduml

```



4.4 在模板程序基础上，增加其他敌机类和道具类

请根据设计的 UML 类图编写代码，添加相应的敌机类和道具类。

本次实验提交版本需完成以下功能：

- ✓ 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
- ✓ 精英敌机坠毁后随机产生某种道具（或不产生道具）
- ✓ 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无需具体实现，生效时只需打印“FireSupply active!”“BombSupply active!”语句即可。

5. 实验提交

提交方式：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

提交物：项目代码

包含：

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