实验二报告

# 单例模式

1. 应用场景分析

*描述飞机大战游戏中哪个应用场景需要用到此模式，目前代码实现中存在的问题及使用该模式的优势。*

创建英雄机时需要用到此模式，并且，英雄机有且仅有一个，一旦被摧毁则游戏结束，因此在游戏中只能实例化一次英雄机。

问题1：目前在Game类中通过new的方式创建英雄机, 违反了设计原则中的单一职责原则。

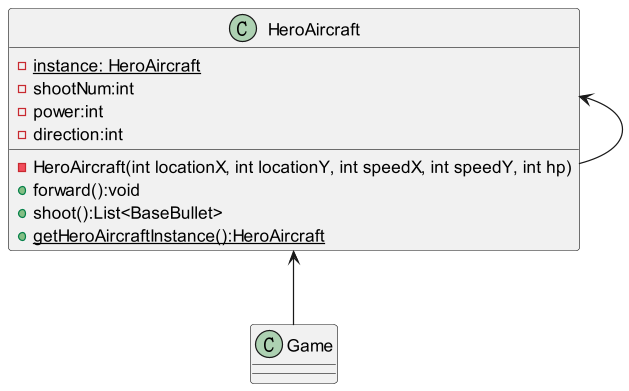
问题2：不能保证英雄机的唯一性，因为外部程序可以用new的方式创造多个实例。

优势：1.保证了英雄机的唯一性，2.遵守了单一职责原则，将对象的创建和使用分离，3.降低了代码的耦合度。

1. 解决方案

*借鉴单例模式的解题思路，设计解决该场景问题的方案。*

1. *将PlantUML插件绘制的类图截图到此处*
2. *描述你设计的UML类图中的每个角色（类、接口），并对它的关键属性、方法和作用进行简要说明。*



Class HeroAircraft， 是英雄机的类，关键属性是private static 的**instanc**e，关键方法是public static synchronized的**getHeroAircraftInstance**方法， 其作用是通过懒汉式的方法创建英雄机并且保证英雄机只会存在一个实例。

在Game中实现游戏的主要逻辑，要实例化英雄机的时候需要调用getHeroAircraftInstance方法来创建HeroAircraft的实例，保证实例只有一个。

# 工厂模式

1. 应用场景分析

*描述飞机大战游戏中哪个应用场景需要用到此模式，目前代码实现中存在的问题及使用该模式的优势。*

在游戏中，敌机会源源不断从界面上方产生，需要生成Mob，Elite，Boss（暂未实现）三种敌机，敌机可以被英雄机击毁后消失，敌机也能对英雄机造成伤害。Elite，boss被击毁的时候有概率掉落3种道具中的其中一种，道具和英雄机碰撞时生效。

问题1：没有对Game类隐藏生成敌机or道具的细节,违反单一职责原则。

问题2：违反了开闭原则，因为不方便增加新类型的敌机和道具。

问题3：违反依赖倒转原则，因为创建过程面向具体实现而不是接口，如果想要增删敌机的属性，要在所有的new运算中修改。

使用工厂模式的优势：

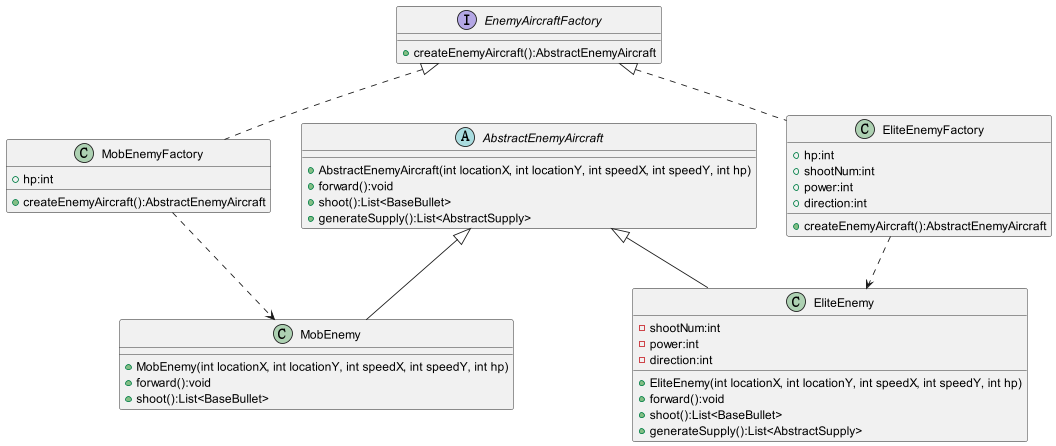
1. 符合单一职责原则：对game类隐藏创建细节，降低耦合度。
2. 符合开闭原则：方便增加新的类型的道具和敌机，只需要增加新的工厂累。
3. 符合依赖倒转原则：创建过程依赖于抽象接口。
4. 解决方案

*借鉴工厂模式的解题思路，设计解决该场景问题的方案。*

*提示： 敌机和道具分开画两张图。*

1. *将PlantUML插件绘制的类图截图到此处*
2. *描述你设计的UML类图中的每个角色（类、接口），并对它的关键属性、方法和作用进行简要说明。*

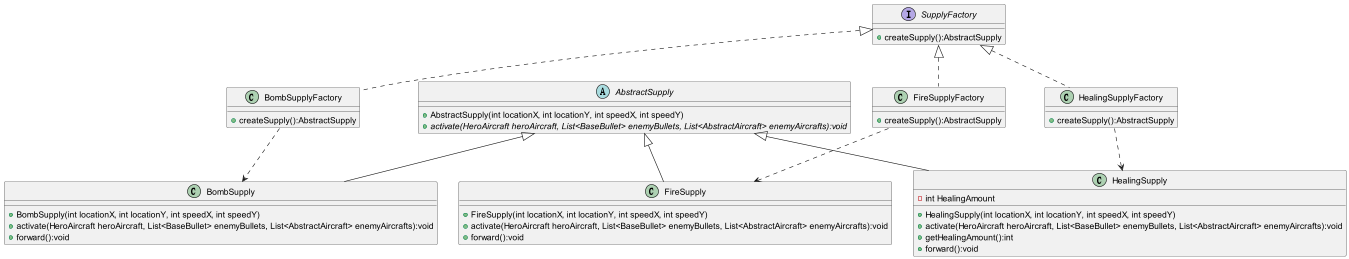
1.工厂模式创建敌机：



AbstractEnemyAircraft类是抽象敌机, EliteEnemy是精英敌机类, MobEnemy是普通敌机, 他们继承自AbstractEnemyAircraft类。EnemyAircraftFactory是敌机的工厂接口, EliteEnemyFactory精英敌机工厂类和MobEnemyFactory普通敌机工厂类实现了EnemyAircraftFactory工厂接口。

关键方法createEnemyAircraft, 作用是生产敌机。关键属性是hp, shootNum, power, direction,这些是敌机的属性。其他方法还有forward, shoot, generateSupply

2.工厂模式创建道具



AbstractSupply类是抽象道具, BombSupply是炸弹道具, HealingSupply是加血道具, FireSupply是火力道具, 他们继承自AbstractSupply抽象道具类。SupplyFactory是道具的工厂接口, BombSupplyFactory炸弹道具工厂类和HealingSupplyFactory加血道具工厂类和FireSupplyFactory火力道具工厂类实现了SupplyFactory接口。

关键方法是createSupply, 作用是生产道具, 以及active, 作用是使道具生效; forward方法的作用是使elite敌机被击毁产生的道具沿elite敌机的飞行方向纵向飞行, 但速度变慢为5。

HealingSupply还有属性HealingAmount以及公共方法getHealingAmount， 作用分别是给英雄机回血的血量以及得到这个量值。