

实验四报告

一、策略模式

1. 应用场景分析

应用策略模式的场景是飞机完成射击功能以及敌机坠毁时产生道具两种场景，在飞机大战游戏中，英雄机和各种类型的敌机所发射子弹的图片、子弹数量、火力值和弹道都不相同。普通敌机不发射子弹，英雄机、精英敌机和 Boss 敌机按照各自方式发射子弹。且火力道具生效后英雄机弹道会发生改变。故可以使用策略模式。精英敌机和 Boss 敌机坠毁时产生道具数量不同，产生方式不同。采用策略模式可以提高代码修改的效率。

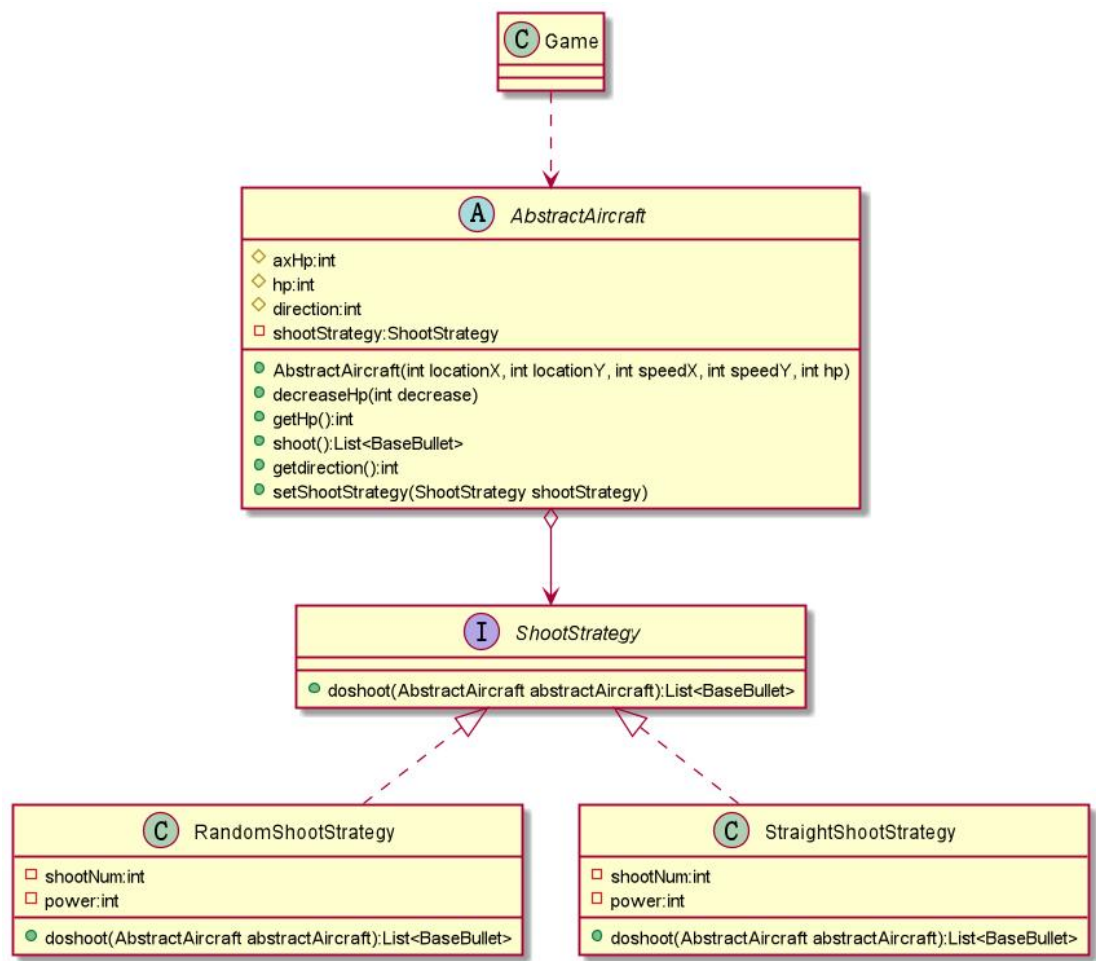
目前射出子弹的代码均写在英雄机、精英敌机和 Boss 敌机类的内部，产生道具的代码写在精英敌机和 Boss 敌机类的内部，如果发射方式或产生道具方式需要替换，需要更改代码。效率很低，且不符合开闭原则。

2. 解决方案

解决方案：

运用策略模式的思想，将射击的算法封装成一个一个类，实现同一个射击方法接口，产生道具的算法封装成一个一个类，实现同一个产生道具方法的接口。所有飞机的父类 `AbstractAircraft` 使用射击方法接口完成射击功能。所有敌机的父类 `AbstractEnemy` 使用产生道具方法完成产生道具功能。飞机子类中只需设置相应的策略。

射击方法的 UML 图：



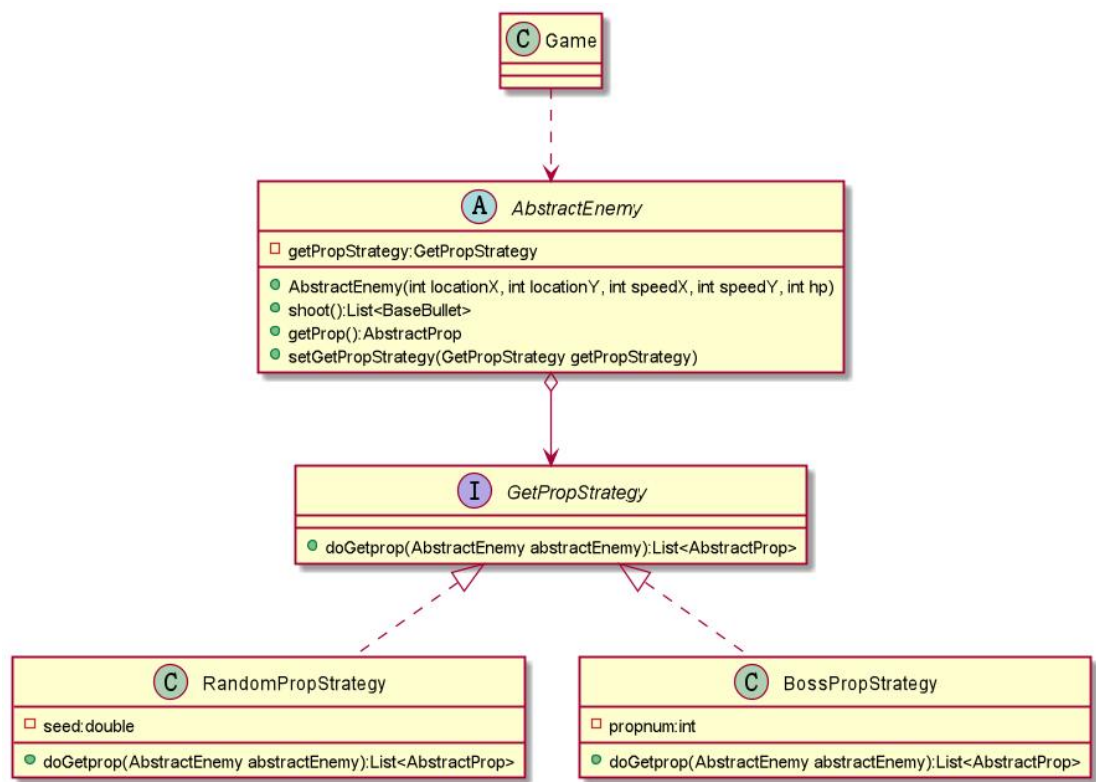
描述：

图中 ShootStrategy 为射击策略接口，RandomShootStrategy 为散射策略类，StraightShootStrategy 为直射策略类，内部定义了 int 型变量 shootNum 和 power，表示单次射击子弹数，和子弹伤害。通过 doshoot 函数产生子弹集合完成对应的子弹射击功能。

AbstractAircraft 作为环境类使用射击策略，AbstractAircraft 内部设定 int 类型的变量 direction 和 ShootStrategy 类型的变量 shootStrategy。通过 setShootStrategy 方法设置飞机所使用的射击策略。通过 shoot 方法调用射击方法的 doshoot 方法获得子弹集合，doshoot 通过 direction 判断射击方向，从而辨别英雄机或敌机，使用对应子弹图片。火力道具只需通过 setShootStrategy 方法更改英雄机所使用的射击策略达到更改弹道的效果。

Game 类作为游戏端呈现效果。

产生道具方法的 UML 图：



描述：

图中 **GetPropStrategy** 为产生道具策略接口，**RandomPropStrategy** 为随机产生单个道具的策略类，内部定义随机数种子 **seed**，**StraightShootStrategy** 为 Boss 机坠毁产生道具策略类，内部定义 **propnum** 表示一次产生道具数量。通过 **doGetprop** 方法产生道具集合完成对应的生成道具功能。

由于敌机坠毁才产生道具，**AbstractEnemy** 作为环境类使用产生道具策略，内部设定 **GetPropStrategy** 类型的变量 **getPropStrategy**。通过 **setGetPropStrategy** 方法设置飞机所使用的射击策略。进而通过 **getProp** 方法调用对应策略的 **doGetProp** 方法得到道具集合。

Game 类作为游戏端呈现效果。

二、数据访问对象模式

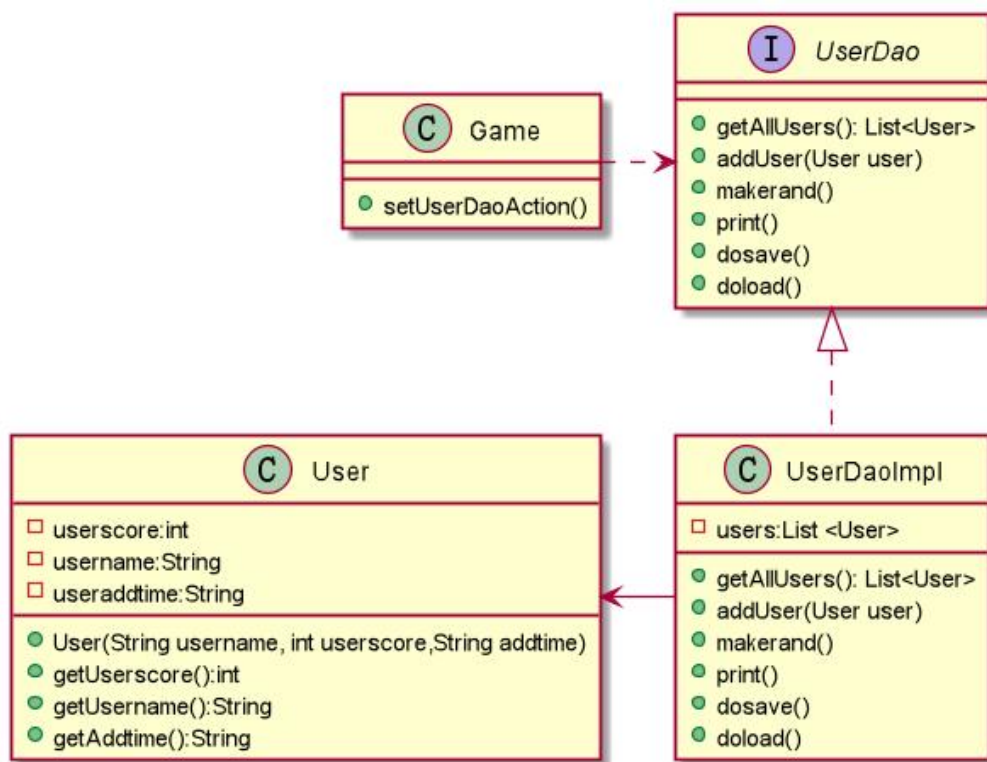
1. 应用场景分析

飞机大战中每局游戏需要记录英雄机得分形成用户数据，用户数据内容包括名次、玩家名、得分和记录时间，游戏结束打印出得分排行榜。故而可以应用数据访问对象模式完成对数据的访问，实现该功能。

2. 解决方案

解决方案：采用数据访问对象模式的思想，将用户数据看作一个类，建立一个数据访问对象接口，在建立一个实现接口的实体类处理用户数据。

UML 图：



描述：

`UserDao` 是数据访问对象接口，`UserDaoImpl` 是实现接口的具体类，`User` 则是用户类，`UserDao` 中设定 `getAllUsers` 方法用于得到用户数据集合，`addUser` 方法用于添加用户，`makerand` 方法用于对用户按照分数排序，`print` 方法打印得分排行榜，`dosave` 方法将用户数据存入文本，`doload` 方法从文本读入数据存入 `User` 数组集合。`UserDaoImpl` 定义模拟数据库数据的 `User` 类型链表 `users`，并实现接口的方法。

`User` 类中定义用户数据，玩家名 `username`、得分 `userscore` 和记录时间 `addtime`，名次用在链表所处位置记录。并设置 `getUserscore`, `getUsername`, `getAddtime` 方法可以分别获取用户相应数据。

`Game` 通过 `setUserDaoAction()` 实现对数据库的应用。