实验四报告

11班 200111132 吴桐

# 策略模式

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。*

*现在飞机的射击子弹的方式仍然是和飞机对象绑定的，让飞机（无论是自己还是敌机）在中途改变设计方式较为困难*

*策略模式将责任与算法分开，射击是统一的动作，但是我们需要不同的算法返回不同飞行方式的子弹*

*所以我们利用策略模式来设计射击子弹，给飞机添加一个策略的接口，在创造实例，或者是游戏过程中接受道具改变方式，只需要更换飞机的策略实例即可。*

*射击策略主要利用飞机自身自带的属性，传入参数，策略的射击函数方法返回子弹序列。*

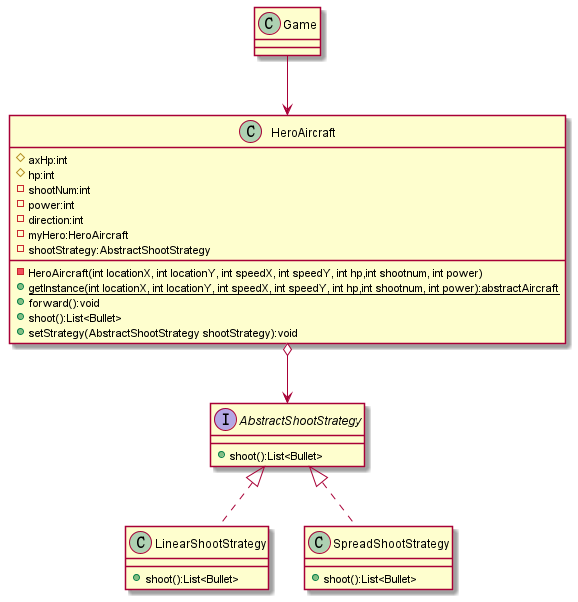
*这里我设计了直线射击和散射两种最基础的（后期再慢慢加入奇奇怪怪的射击方法）*

1. 解决方案

*借鉴策略模式的解题思路，设计解决该场景问题的方案。结合飞机大战实例，绘制具体的UML类图。*

1. *将PlantUML插件绘制的类图截图到此处（随代码也提交一份）*
2. *描述你设计的UML类图结构中每个角色的作用，并指出它关键的属性和方法。*

*a.下一页下一页，puml图在下一页*



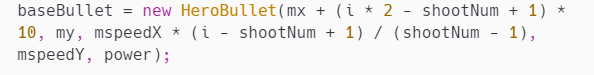
*b.详细*

*这里英雄机添加了策略接口的属性，还有设置策略的方法。*

*射击这里自己重构代码的时候为了避免策略为空，但是保留了默认射击的策略，如果策略不为空，则调用策略的射击函数*



直射和原来差距不大，散射的射击函数主要依靠根据子弹射出位置线性改变子弹x方向上的速度不一样，来造成散射的效果



# 数据访问对象模式

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，分析使用该模式的优势。*

*后续我们需要为每个玩游戏的人保存他的游玩数据，并且我们需要打印结果，游戏需要排行榜！*

*但是目前游戏并没有加入保存游戏数据的能力，所以我们利用数据访问对象模式*

*有一个类是我们需要存储的数据（一切事物对象化），另一个接口是我们需要操作数据的方法。*

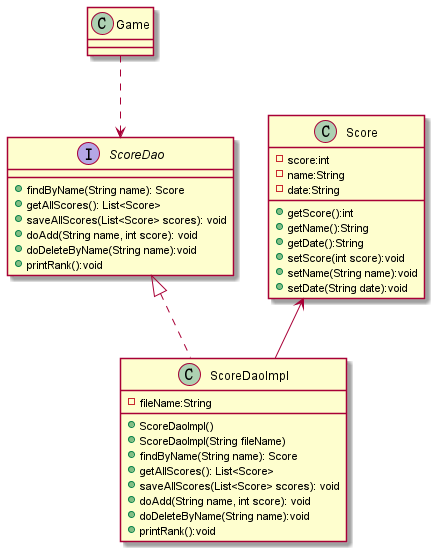
*操作数据实际方法的实现可以有很多种，通过具体的类来规定合适的算法*

*目前没有规划说用哪种数据库驱动，那就直接写入txt文件吧*

1. 解决方案

*借鉴数据访问对象模式的解题思路，设计解决该场景问题的方案。结合飞机大战实例，绘制具体的UML类图。*

1. *将PlantUML插件绘制的类图截图到此处（随代码也提交一份）*
2. *描述你设计的UML类图结构中每个角色的作用，并指出它关键的属性和方法。*



*Score 类*

*属性分别为得分，玩家名字，日期*

*方法为标准的构造方法*

*ScoreDao接口*

*写入文件主要是通过写入对象的方法实现（学习一下序列化）*

*findByName 通过名字查询返回当前排行榜的得分对象*

*getAllScores 返回一个列表，元素为所有当前排行榜的对象*

*saveAllScores 传入一个列表，将列表中的元素全部写入文件*

*doAdd 传入名字和得分参数，插入排行榜记录文件*

*doDeleteByName 传入名字参数，删除该名字下的记录*

*printRank 按名次打印当前排行榜中所有得分记录*

*成绩打印效果如下：*

