

西南科技大学

Southwest University of Science and Technology

设计模式实验报告

题目名称：基于 OpenCV 的炸弹类小游戏代
码重构

小组成员	袁海 7220200622
------	---------------

	罗月 7220200619
--	---------------

	田杰 7220200635
--	---------------

	周隆放 7320200029
--	----------------

二〇二〇年十二月

西南科技大学

基于 OpenCV 的炸弹类小游戏代码重构

摘要：

代码的逻辑往往很复杂。通常在代码运行大规模的工作的时候，不恰当的设计方式往往会增加重用的代码、资源竞争的问题。本课题获取到基于 OpenCV 的面向对象的炸弹类小游戏源码，通过设计模式重新设计对象之间的依赖关系以及增加新的功能，从而使每个对象的分工更加明确和减少内存的开销。本课题通过代码的重构进而分析设计模式优越性。

关键词：设计模式；面向对象；OpenCV；运行效率

OpenCV – based bomb game code refactoring

Abstract:

The logic of the code is often complex. Often when code is running on a large scale, improper design can lead to increased reuse of code and competition for resources. This topic obtained based on OpenCV object-oriented bombs small game source code, through the design pattern to redesign the dependency between the objects and add new functions, so that each object more clear division of labor and reduce the memory overhead. This paper analyzes the design pattern optimality through code reconstruction.

Key words: Design patterns; Object oriented; OpenCV; Operation efficiency

目录

- 一 . 引言.....5
 - 1.1 源码介绍.....5
 - 1.2 存在的问题.....5
 - 1.3 重构目的.....5
- 二 . 项目初始化简介.....5
 - 2.1 项目界面与功能介绍.....5
 - 2.2 代码分析.....6
- 三 . 项目重构与分析.....8
 - 3.1 单例模式.....8
 - 3.2 工厂模式.....8
 - 3.3 抽象工厂模式.....9
 - 3.4 建造者模式.....11
 - 3.5 组合模式.....12
 - 3.6 装饰模式.....13
 - 3.7 观察者模式.....15
 - 3.8 策略模式.....15
- 四 . 总结.....16
- 五 . 致谢.....16
- 六 . 参考文献.....17

一 . 引言

1.1 源码介绍

此课题的源码是组内同学以前写过的 C++面向对象的代码（代码行数在 2000 行左右），是基于 OpenCV 的开发的一款图形化界面的炸弹类小游戏，属于面向对象编程，符合此课题的运用设计模式来进行重构的需求。

1.2 存在的问题

代码里存在的问题。首先，主角类与敌人类所包含的字段和函数高度重合，造成代码的冗余。其次，所有类都依赖于资源类，这使耦合性过高，不利于后续的开发。

1.3 重构目的

本课题的目的是通过设计模式的手段重新整理代码里类与类之间的关系，让运行的逻辑更加规范化，使代码有更好的扩展性，减少代码之间的耦合性，从而增加代码的健壮性和鲁棒性。

二 . 项目初始化简介

2.1 项目界面与功能介绍

游戏初始化可以选择不同的地图场景，如图 1 所示。载入地图进入特定的场景后会生成两个玩家如图 2 所示。玩家可以在空白的位置有限制地释放炸弹，被炮弹炸开地箱子有一定的几率掉落一定的物品，玩家可以捡起掉落的物品增加自身的属性。游戏的任何角色被炮弹炸到就判定为死亡，另一方就获胜。



图 1

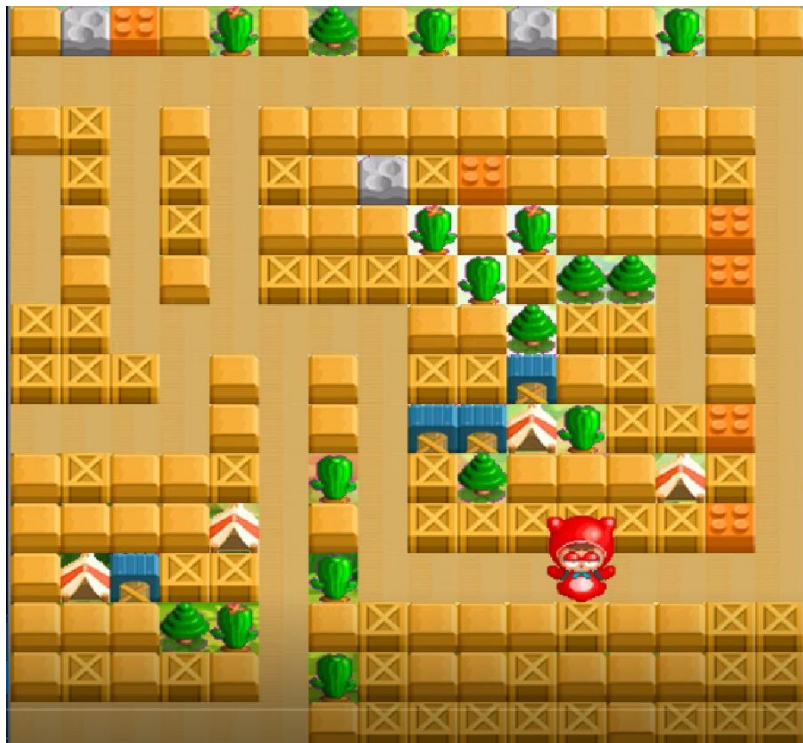


图 2

2.2 代码分析

代码中有游戏引擎类、地图类、资源类、主角类、敌人类、炮弹类。如图 3 和图 3.1 所示。游戏引擎类管理整个游戏的运作，包括资源的调度、地图的生成、角色的生成、游戏行为的产生以及游戏胜负的判定。主角类与地人类负责与地图类和炮弹类的交互，地图类掌握整个地图资源的分配。炮弹类掌握炮弹的具体信息。资源类掌管图片资源，将所有的图片调入内存，方便其他类需要的时候直接取图。但是代码存在一定的问题，比如主角类和敌人类代码相同，存在冗余，代码的可重用性不高。而且资源类掌握所有的类的图片资源，所有的类取图片都依赖于这个这个类，这段代码的关系的耦合性就很高。此外还有部分功能没有实现，

如果实现新的功能，对这样的设计的代码的可扩展性要求就很高。我们将这些类利用一些设计模式给予改进，并且新增添了一些功能，结合恰当的设计模式，从而增加代码的可扩展性。

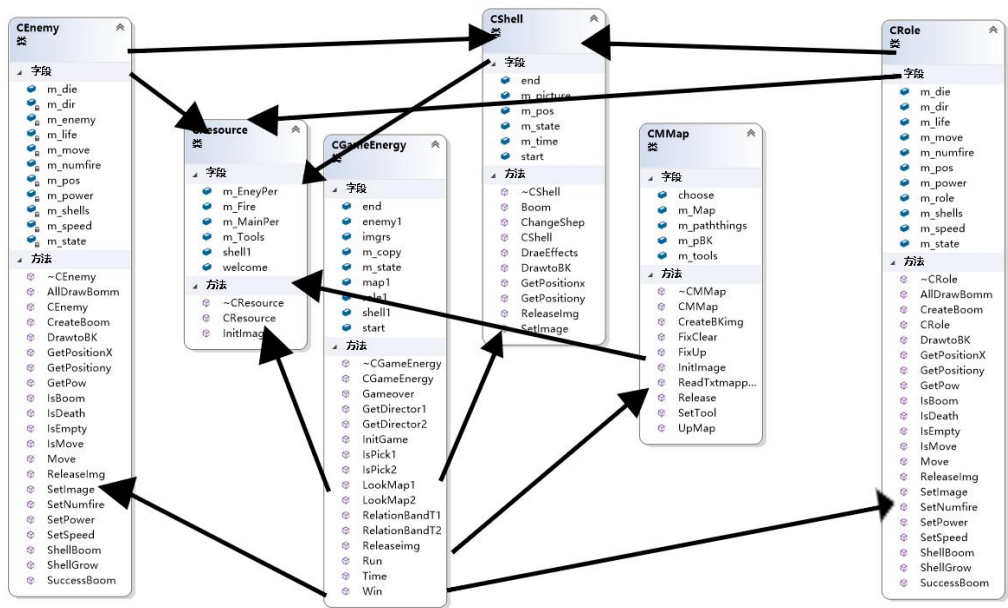


图 3

首页	类	文件
组合类型列表	组合类型索引	组合类型成员
组合类型列表		
这里列出所有类、结构、联合以及接口定义，附带简要说明：		
CEnemy	角色的敌人类	
CGameEnergy	游戏引擎类	
CMap	游戏的地图类	
CResource	资源库类	
CRole	角色中的主角类	
CShell	炮弹类	

图 3.1

三 . 项目重构与分析

3.1 单例模式

我们将单例模式用在地图类上，因为整个游戏运行的过程中地图只有一个，是唯一实例。其操作方法是将地图类的构造函数私有化。如图 4 所示。

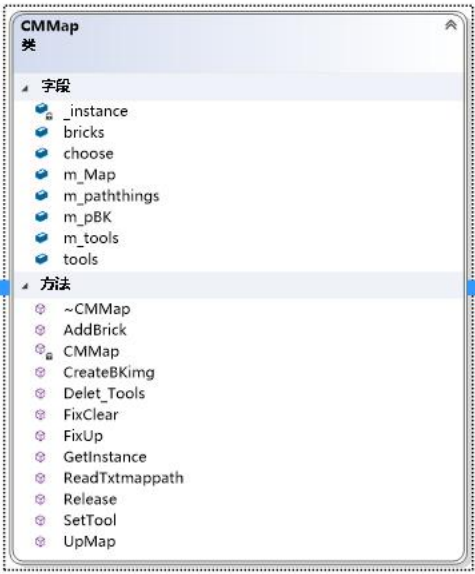


图 4

3.2 工厂模式

角色创建的时候，主角类和敌人类应该都属于角色类，所以我们将角色类派生出主角类和敌人类，调用角色创建工厂来先创建主角类再创建敌人类。如图 5 所示。主程序调用角色生成工厂，工厂封装了创建角色的代码，访问角色基类初始化角色信息再返回角色指针。使用工厂模式的好处是可以不在游戏主函数里写创建角色、初始化角色的代码，将创建角色的流程交给工厂实现。

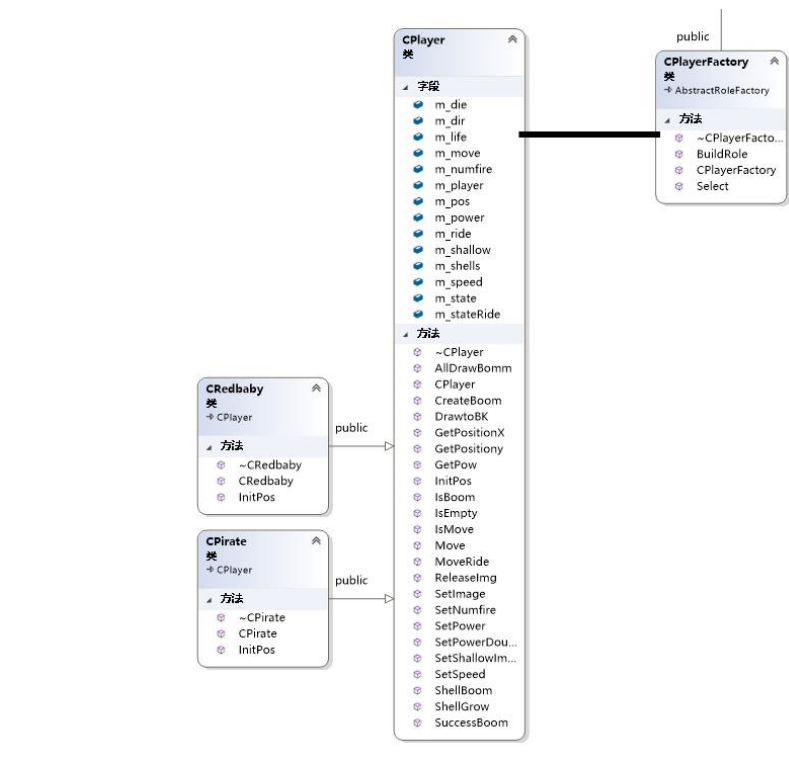


图 5

3.3 抽象工厂模式

我们为角色新添了选择皮肤的功能，用户在创建角色的时候供有四种皮肤可以选择，皮肤的创建我们也是使用了工厂模式来创建如图 6 所示。由于创建角色和创建皮肤各有一个工厂，用户生成角色既要有角色信息又有要角色皮肤，所以我们将这两个工厂封装成了一个抽象工厂如图 7 所示。实现的界面如图 8 所示。当我们的程序创建角色的时候，角色信息初始化与角色的皮肤是同一个产品的不同的两个方面，两个维度都各自有多样化的选择，抽象工厂模式就很好的适用了当前的问题，使客户端只关心创建角色的工厂，而不关心工厂下还有多个子工厂协同工作。

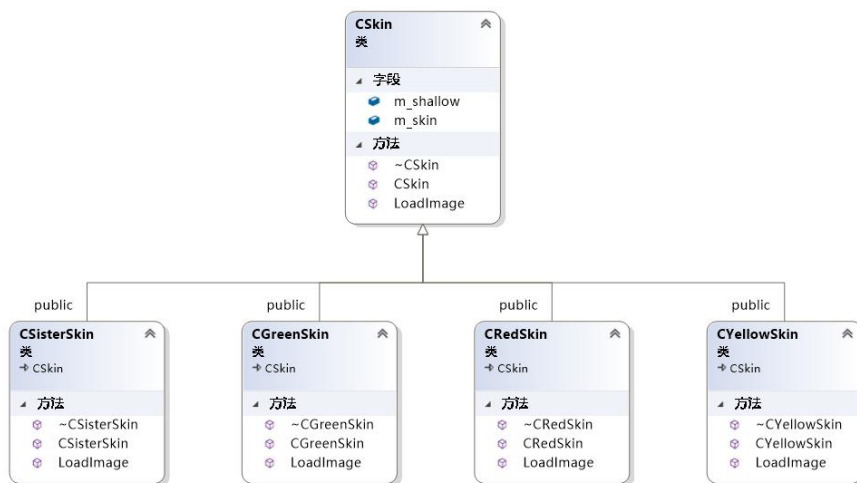


图 6

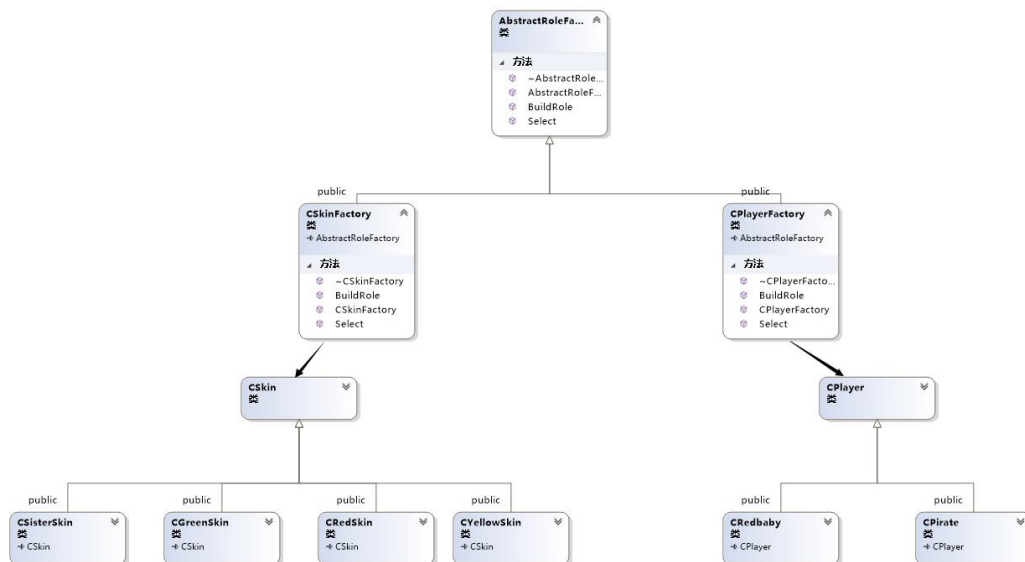


图 7

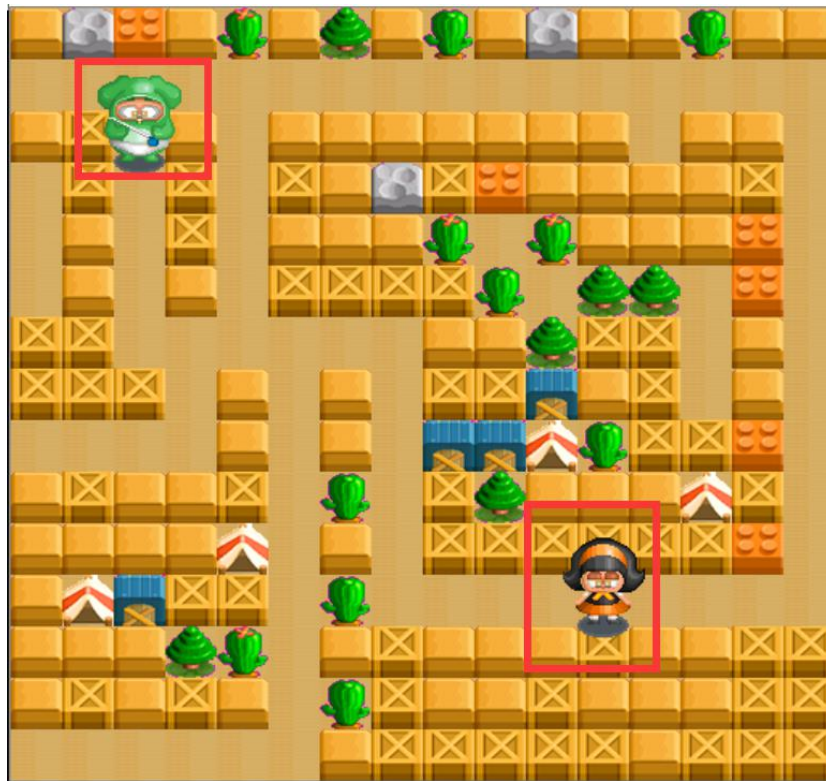


图 8

3.4 建造者模式

按照更加丰富的游戏需求，我们需要为角色创建能有坐骑的功能，我们是将角色类和坐骑类组合成了一个有坐骑的角色类，因此我们采用了建造者模式。如图 9 所示。我们将坐骑的作为基类派生出 UFO 类，再构建一个 CRideBuilder 这类，这个类里面封装了建造坐骑的固定步骤。实现界面如图 10 所示。我们将构建坐骑的过程与最后坐骑的表示愤慨，可以降低程序的耦合性。

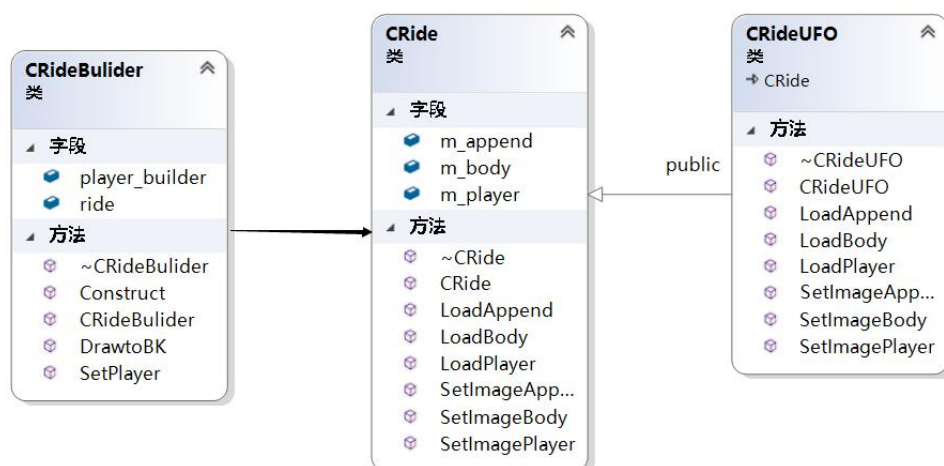


图 9



图 10

3.5 组合模式

用户游戏体验的时候出现了一些问题，即建筑物底色是空白容易受到其他特效的影响。我们将每一个建筑物都设定成一个底色地砖，再在这个地砖上多样化地生成建筑物这样就解决了绘制的问题。此时组合模式就很好的解决了这样的问题如图 11 所示。地砖类都有一个底色图片，同时还有一个 vector 用于存放建筑物图片作为他的子类如图 12 所示。我们利用地砖这种存在树形结构的模式设计成组合模式就解决了游戏中出现的问题。

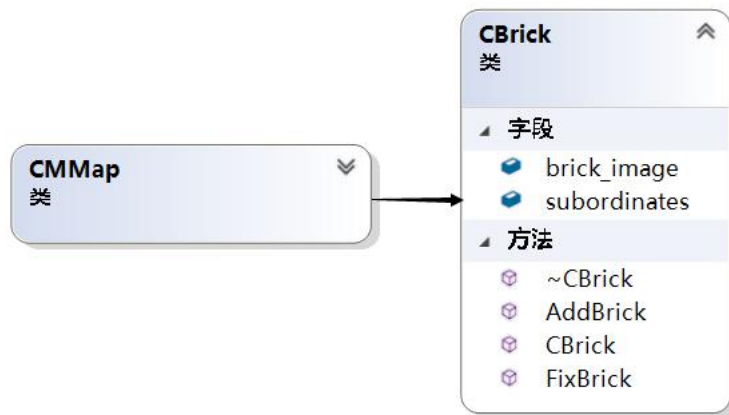


图 11

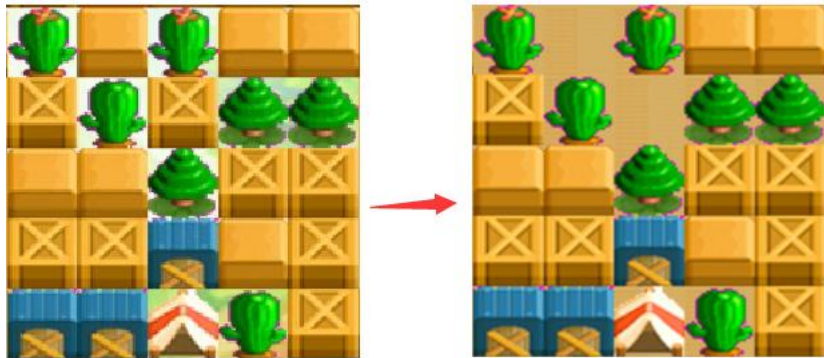


图 12

3.6 装饰模式

游戏的功能之一是能掉落道具，供玩家捡取。由于游戏的需求增加，会要求更多种的道具出现，为了满足想要某一样道具拥有额外的功能，装饰者模式提供了较好的解决方案。我们将道具作为基类抽象出鞋类、药水类、气球类。如图 13 所示。此时我们给药水类增加爆炸威力的功能如图 14 所示。实现方法是将药水类的指针赋值给装饰类，再在装饰类的子类紫色药水类新增增加威力的功能。由于需求的不断变更，装饰类和被装饰类可以独立扩展不会耦合。

3.7 观察者模式

游戏评判输赢的机制取决于炸弹爆炸是否伤害到角色，所以我们利用观察者模式为每一个生成的炸弹创建一个观察者，等炸弹爆炸的时候观察炸弹爆炸的距离和玩家的位置来判定玩家是否死亡。如图 15 所示。我们将观察者作为基类派生出观测伤害的观察者，目标对象是炸弹，每一个观察者用统一的 vector 存起来，炸弹爆炸观察者执行完功能后也将删除。如图 16 所示

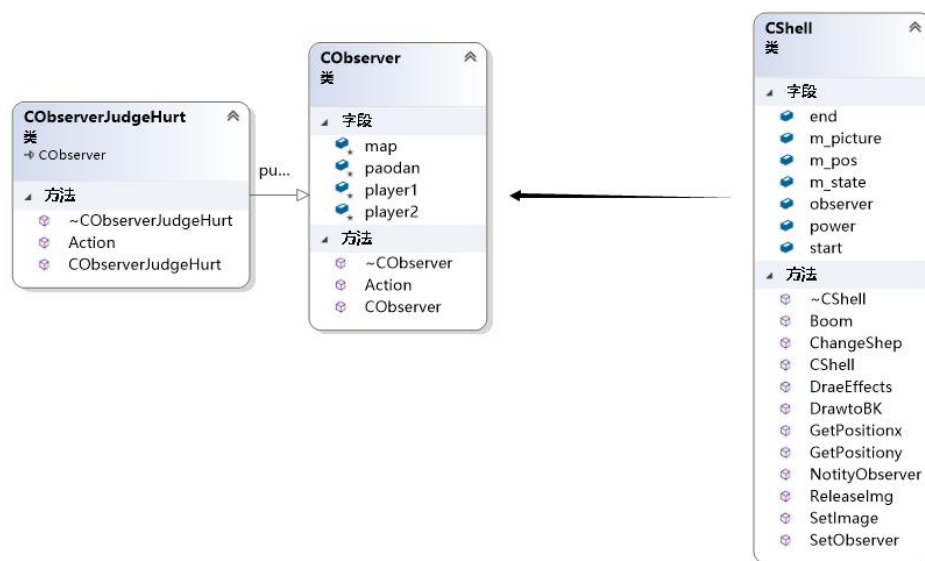


图 15



图 16

3.8 策略模式

由于需求的增加，炸弹外观的但单一需要改进。因此我们用了策略模式，使程序运行的过程的过程中动态的炸弹的外观。我们将选择炸弹的外观当成一种策

略。生成一个策略基类，派生出四种策略子类供选择如图 17 所示。生成炮弹的时候系统随机决定应用哪一种外观策略，如图 18 所示。

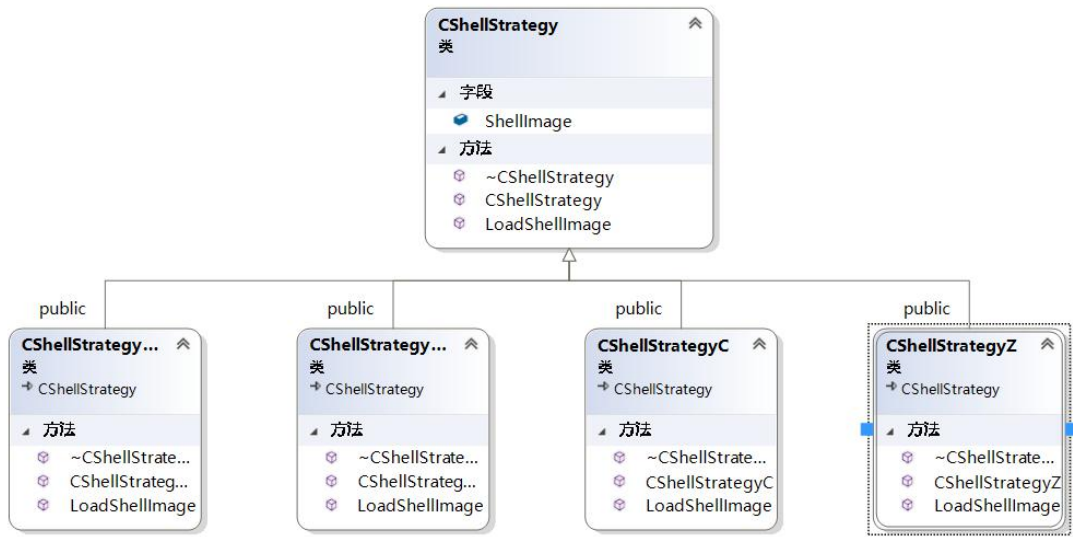


图 17

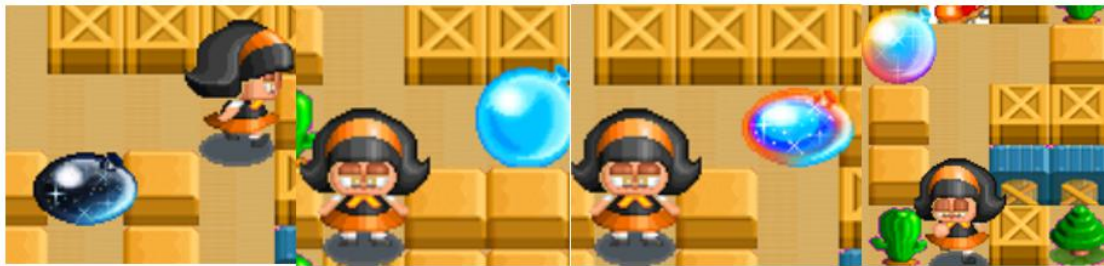


图 18

四．总结

在这几周的代码重构中，我们深刻了解了设计模式的思想，并且在实际的应用上感受到了合适的模式对代码扩展的方便性，能降低耦合且便于用户理解与修改。同时我们也学习到了写代码的规范，尽量不要让函数行数长且冗余，应该积极思考其运转的逻辑思维，找寻合适的方法将其分开。我们组对设计模式的理解的深入也难免带来困惑：比如策略者模式有相似于工厂模式，在使用的时候难区分；工厂模式也有点像建造者模式等等。在享受愉快的学习过程中也因为时间紧的缘故很多问题还没想明白就匆匆结束了。希望以后的能有机会再来解开这样的困惑。

五．致谢

感谢陈念年老师的授课与解惑，带领我们学习重构的思想，也感谢老师的耐心与细致使我们在上课的时候能慢慢理解设计模式的思想精髓。同时感谢实验室

的学长学姐在我们组碰上困难的时候给予帮助,是我们在做作业的时候度过难关。感谢一起做作业的同学给我们的提供新颖的想法。感谢遇见这么多人给予我们帮助。

六 . 参考文献

- [1].fErichGamma, 加马, 李英军. 设计模式:可复用面向对象软件的基础[M]. 机械工业出版社, 2000.
- [2]. BrunoR.Preiss. 数据结构与算法:面向对象的 C++设计模式[M]. 电子工业出版社, 2001.
- [3]. 李智勇. 深入浅出设计模式[J]. 程序员:CSDN 开发高手, 2004(9):48-50.
- [4].ErichGamma, 加马, 李英军. 设计模式:可复用面向对象软件的基础[J]. 2000.
- [5]. 夏伦先, 唐胜群, 马捷,等. 使用设计模式指导 UML 类模型的建立[J]. 小型微型计算机系统, 2002(04):77-80.