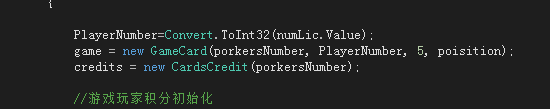
1. 设计思路

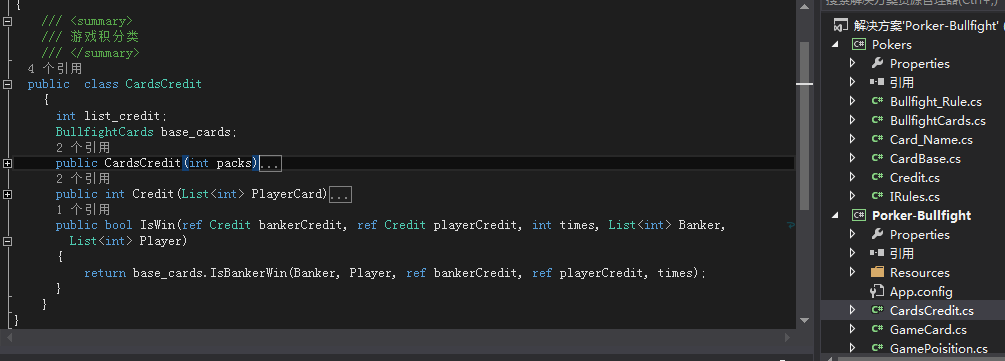
主要功能是由该类库实现的，分为规则接口、规则类、牌基类，斗牛游戏类实现规则接口并继承了牌基类。综合积分类，实现游戏的主体功能。但是Card\_Name既在规则类中实现，又在主类中实现，嵌套的调用了setRules方法。导致在调用游戏胜败当中又重新实例化了一个游戏类，很麻烦而且容易出错，特别是对于牌的副数。

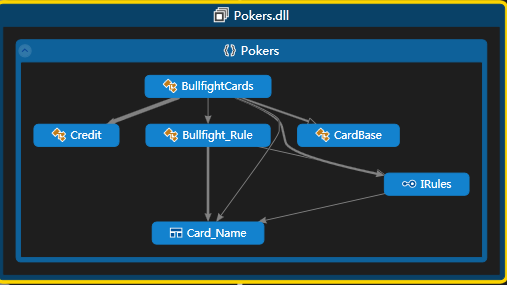
1. 所以游戏类和规则类的使用设计不明确。应该是规则类在上层，游戏类在下层，这样在使用规则的时候可以直接根据游戏类初始化，而且对于单一的游戏类也不会出现规则类不一样的情况。比如算牌点数等规则上。不然会出现这种情况。



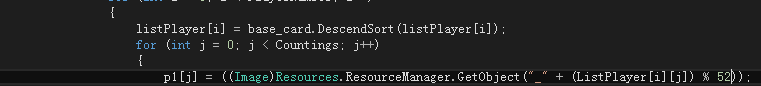
所以游戏类也不应该出现规则+牌基类的情况，应该就只是对牌基类的实现进行修改。牌的其他功能应该留给规则类来实现。

不用实现CardsCredit类了，直接在GameCard类中实现游戏类并初始化规则类就可以了。

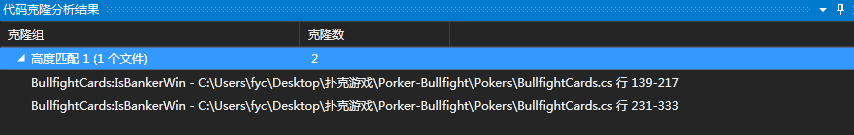




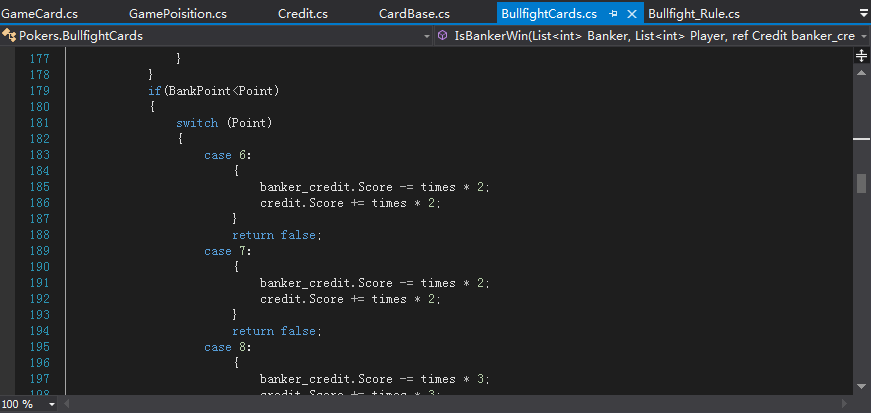
1. 对于小大王的情况采取了硬编码的情况，导致游戏如果不含了小大王则改的地方都有四处。所以对于小大王的情况设计静态变量。



1. 代码克隆冗余的情况

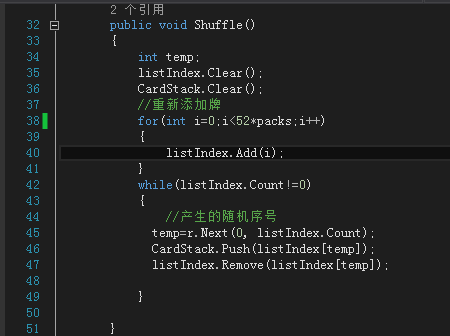


这里主要是因为如果变成计算式子过于死板，可以试着重载操作符，这样的冗余是非常大的。



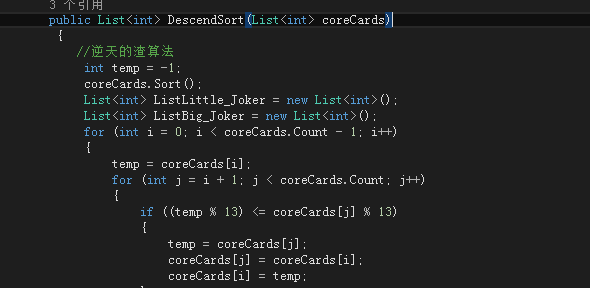
1. 算法的优化

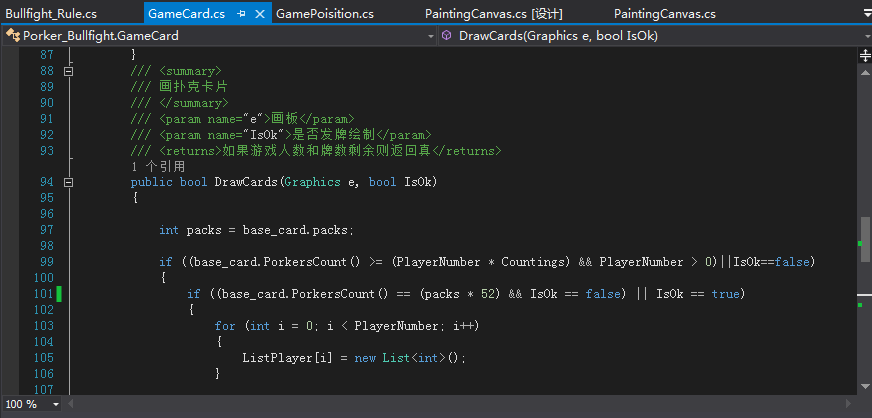
里边有的算法过于渣



发牌类，直接交换次序达到随机目的即可，其实没有必要另外用栈来弹出。

对于扑克排序问题，虽然根据余数可以判断大小，但这里的N^2的平方的排序和大小王的插值问题嵌套过多。可以试试希尔排序等改进一下。



1. 比较好的是采用了迭代进行抽取牌，实现了取多张牌的问题。
2. 用开关量控制了变化窗体的重绘问题
3. 职责不明确，因为必须设计盖牌的绘制，所以必须将发牌和绘制牌分开
4. 。