

第二次课程报告

主要内容、目标及设计思路

主要内容

本次课设主要要求写的是一个坦克大战的游戏，表面上来看是比较复杂的各种对象例如敌方坦克、我方坦克、子弹、道具之间产生联系，但是深层次来看其实就是一些类之间的相互关系而已。

目标

通过利用monospace字体以及window的句柄实现任意位置的字符的打印，和通过不断处理各种类之间的关系，以构造出坦克大战的基本功能实现。

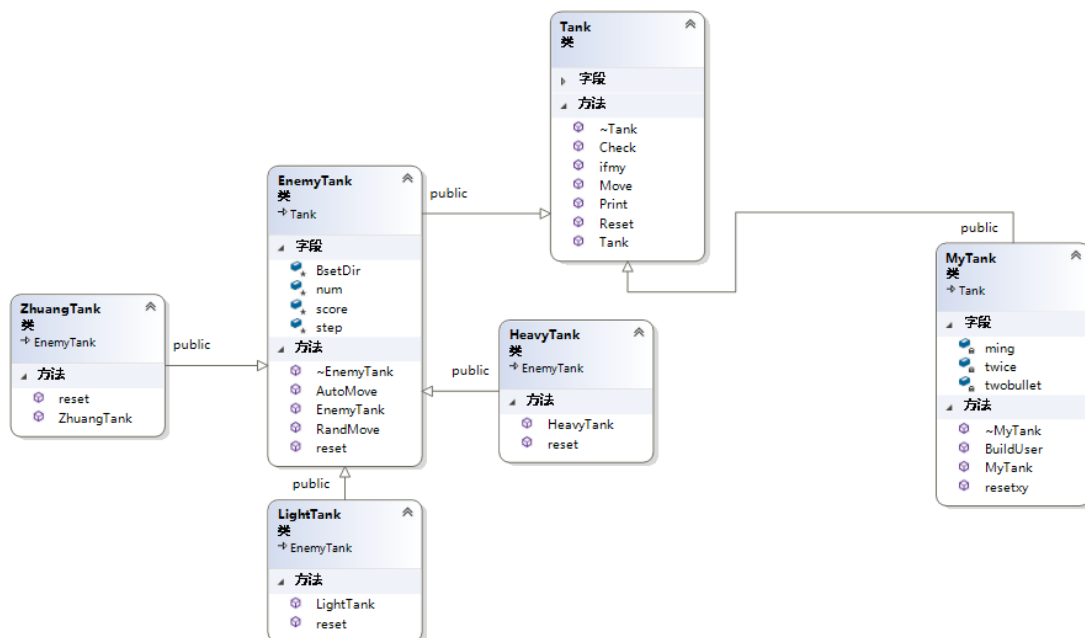
设计思路

初步设想是先设计一个游戏面板类，用于打印各种对象的模样，然后设计一个游戏核心类，用于处理各种类之间的复杂的关系用的；地图是通过一个一个的数字输入来实现的（🗺️）；其余就是各种地图上的对象了，通过继承和多态实现一定的代码复用。

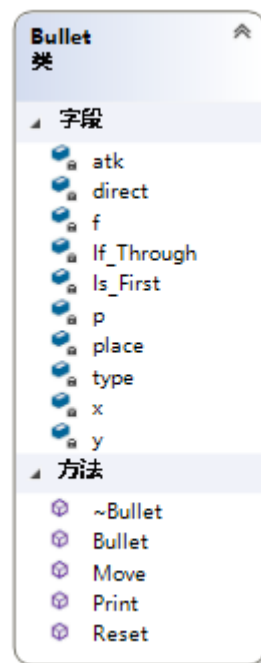
主要类的设计及用到的OOP设计思想

主要类的设计

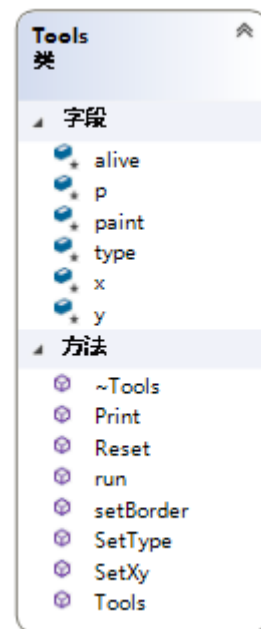
坦克类：



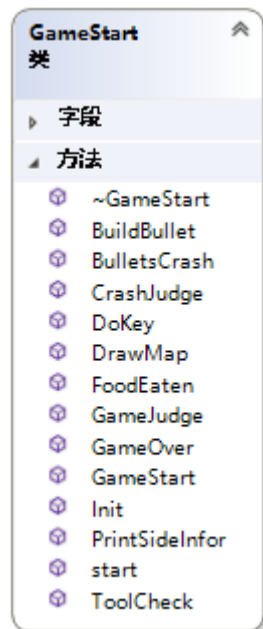
子弹类：



道具类:



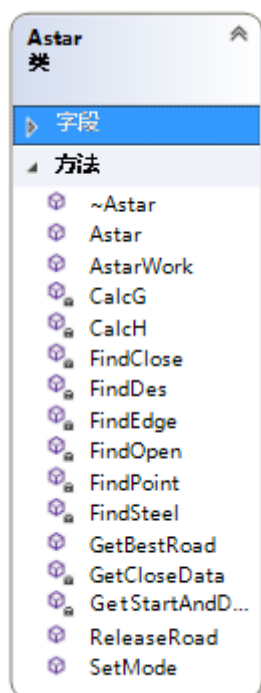
游戏控制类:



实现游戏选关功能的游戏架构类：



用于实现A*算法的AStar类：



用到的OOP思想：

①继承：从刚刚的类图可以看出，由于我的坦克与敌方的坦克均有移动，打印，碰撞的实现，所以我将我的坦克与敌方坦克均继承于基本的坦克类型，而且敌方的坦克有三种不同的类型，又因为每种类型的特征均不相同，故将这三种类型的坦克均继承于敌方坦克类

②多态：在游戏控制类中，我只是在构造函数中加入了敌方坦克的类型，后面在使用过程中都是直接用EnemyTank*的类型对这些不同类型的坦克进行控制，且这些坦克都拥有着不同的reset函数等，故通过多态来实现可以减少代码复用。

程序的功能介绍

初始进入游戏，可以选择关卡，该关成功后进入下一关，直到失败为止，每局结束都会显示本局的成绩，每关之间我的坦克的某些功能可以继续使用。其余道具使用，与老师发的pdf中保持一致。

坦克采用A*算法，能够通过障碍，直接到达家的旁边。

遇到的问题与解决方案

①起先是采用深度搜索来实现坦克的AI功能，但是这样的话会对程序本身的效率造成很大的影响，故采用A*自动寻路算法：我的策略是：G是从起点A移动到指定方格的移动代价，估值H我是用普通曼哈顿法，取当前点x-目标x的绝对值与当前点y-目标点y的绝对值两者的和乘10。

②曾经遇到过有时候坦克的子弹有时能打穿铁块有时候打不穿，后来发现是在重置子弹对象的时候，并没有将其If_Through重新赋值。