

**《面向对象的程序设计》**

**大作业**

**题目：** 坦克大战

物联网工程学院 **学院** 计算机科学与技术 **专业**

学 号： 1033180319

学生姓名： 杜昊

班 级： 计科1803

成 绩：

**2020年 6月**

|  |  |  |  |
| --- | --- | --- | --- |
| **班级** | **姓名** | **学号** | **分工** |
| 计科1803 | 杜昊 | 1033180319 | 类的设计、代码测试 |
| 计科1803 | 孙创昱 | 1033180327 | 类的实现、代码框架建立 |

**一、题目要求及分析**

**1、题目要求**

（1）采用面向对象的思想设计程序

（2）采用C++语言进行设计

（3）注意人机友好性

（a）对用户操作，给出足够的提示信息。

（b）程序具有一定的健壮性，不会因为用户的输入错误引起程序运行错误而中断执行。对输入值的类型、大小范围、字符串的长度等，进行正确性检查，对不合法的输入值进行相应处理，不影响程序运行。

（c）对界面进行美化

**2、要求分析**

**①初步分析:**

采用面向对象的思想设计程序。面向对象的三大特征是封装继承和多态。　封装要求将游戏中的物体抽象得到的数据和行为（或功能）相结合，形成一个有机的整体，也就是将数据与操作数据的源代码进行有机的结合，形成“类”，其中数据和函数都是类的成员。继承继承就是子类继承父类的特征和行为，使得子类对象（实例）具有父类的实例域和方法，或子类从父类继承方法，使得子类具有父类相同的行为。多态同一个行为具有多个不同表现形式或形态的能力。是指一个类实例（对象）的相同方法在不同情形有不同表现形式。根据以上特点，我们在完成作业时候，要将游戏中的地图，坦克，子弹等物体封装成类，并且尽量使用继承，实现代码复用，减少工作量。在合适的地方进行函数的重载，以满足不同的需求。

采用C++语言进行设计。新建.cpp文件，尽量采用C++的语法，并且在合适的时候使用C++的标准模板库STL,和图形库EasyX 解决问题。

注意人机友好性。对用户操作，给出足够的提示信息。可以再开始界面比较明显的地方，加上各个按键的功能说明。对输入值进行正确性检查，对不合法的输入值进行相应处理，不影响程序运行。可以仔细考虑各种情况，避免不合理状态的产生。在类的NameSet函数中，使用assert函数，检查传入参数的合法性。游戏过程中设置语句检查数据的合法性，自动修正。对界面进行美化。使用PhotoShop对网上找到的素材进行处理。注意对主界面和显示数据面板的排版。和各种特效的设计。

**②游戏功能要求：**

经典模式（包含单人和多人模式）：

游戏开始界面：用于说明操作方法，给用户提供选择模式，选择难度的界面。

坦克大战游戏游戏主界面和说明性副界面：主界面上包含多种地形：坦克、道具、子弹等；说明副界面上显示关卡数、剩余敌人数、玩家剩余血量。

地形：草丛：坦克在底下移动的时候有隐身效果；水流：坦克不能在水面移动，但是子弹可以在水面移动；钢块：普通坦克打不动，在坦克四星之后可以打动、砖块普通弹可以破坏。

坦克：分为玩家坦克和敌人坦克。玩家坦克有四种形态，后三种分别有子弹速度，子弹数量，子弹威力的增幅。敌人坦克有四种，分别有子弹速度，移动速度，血量的差异。坦克只有在子弹爆炸后才能再次发射。此外，为了防止坦克贴墙造成短时间内造成大量破坏，我们加入子弹冷却时间的限制保证在单位时间内子弹不会有过大的输出。

子弹：可以在地图上直线移动，根据发射者的不同的威力，可以对地图造成破坏，也可以降低坦克血量。

道具：星星道具（star）：可以提高玩家坦克的星级；头盔道具（hat）：可以给玩家提供一段时间的保护罩；手雷道具（bomb）：可以消灭地图上存在的所有坦克；钟表道具（clock）：可以使地图上的敌人静止一段时间。铲子道具（shovel）：可以在一段时间内加强总部的防护壳。坦克道具（tank）：可以给玩家增加生命值。

动态特效：1.坦克生成时时有闪烁效果。2.携带道具的坦克会闪烁，并且带有江南大学的校徽。3.敌方的装甲坦克三条血分别对应不同的颜色。4坦克的履带移动时会动。5.子弹爆炸特效。

娱乐模式：

新增以下素材和道具：

病毒：可以在地图上随机移动。

胶囊：可以在地图上直线移动并消灭病毒。

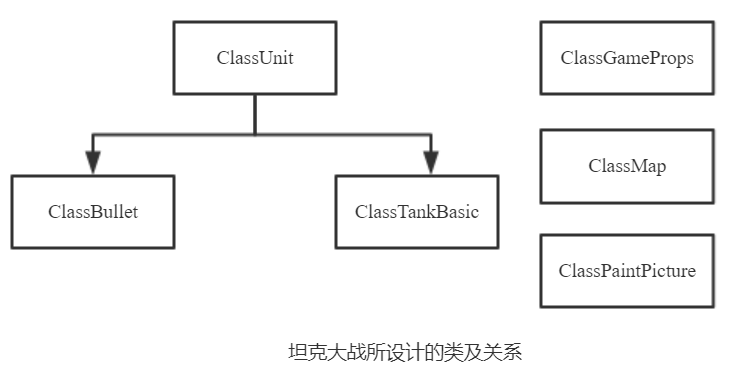
药箱：被放置后碰到病毒后触发，消灭病毒后消失。

救护车：有两种形态，都可以发射胶囊，对冠状病毒造成伤害。第二种形态可以在地上放置药箱。

卡通头像：抵御病毒入侵时的幸存者，如果全部消失游戏结束。

**二、设计思路**

设计思路：坦克大战中主要有地图类、坦克类、子弹类和道具类。为了便于管理，提升代码可重用性，我们设计一个单元类，这样坦克类和子弹类的一部分属性可以共用。除此以外，我们还将绘图功能封装为类。



**1、单元类（ClassUnit）**

**成员变量：**

①绘图坐标mappos，类型为自定义结构体PosI；

②单元类型type，类型为枚举类型UnitType；

③运动方向direction，类型为枚举类型Direction；

④移动速度speed，类型为枚举类型Speed。

**成员函数：**

①构造函数；

②虚析构函数；

③移动判断函数CanMoveForward——判断单元移动是否可行；

④绘图坐标设定函数；

⑤绘图坐标返回函数；

⑥单元类型设定函数；

⑦单元类型返回函数；

⑧运动方向设定函数；

⑨运动方向返回函数；

⑩移动速度设定函数；

移动速度返回函数；

**其他函数：**

①坐标转换函数ToPosI：

该函数的作用是讲绘图坐标转换成地图上基本块的坐标。根据方向和绘图坐标，通过计算，得到改单位在地形图中的整格子坐标，用于单位之间碰撞的判断。

具体实现：如果坦克方向向上且当前纵坐标为不为绘图最小块边长的整数倍时，需要向y轴正方向移动至上一个坐标点。同样如果坦克方向向左且当前横坐标为不为绘图最小块的整数倍时，需要向x轴正方向移动至左边一个坐标点。其他情况下利用除法的四舍五入特性，坦克的坐标除以最小块边长即为对应的地图坐标。

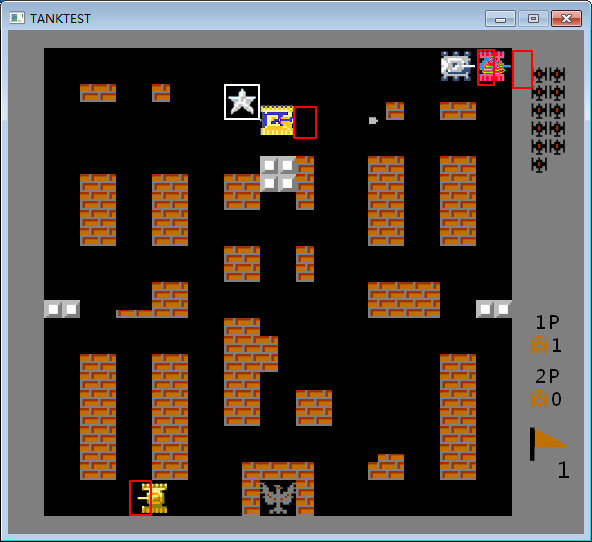
②地形判断函数invalid：

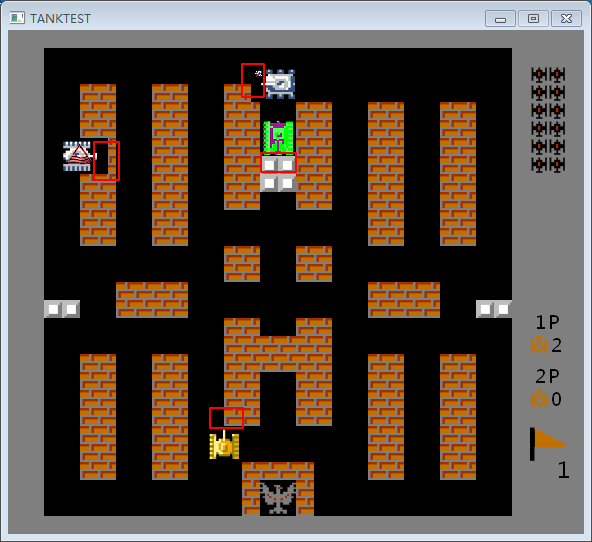
子弹和坦克的判断基本相同。如果该处是砖墙、防爆门、地图边界、总部等返回不可行，否则可行。唯一不同点在于如果被判断的单位是子弹，那么子弹可以穿过海面，而坦克不行。

**部分成员函数实现：**

①移动判断函数：

按照移动函数的坐标对齐方式计算出行列坐标。选择两个坐标点调用地形判断函数判断是否可行，有一个不可行则返回不可行。

****

****

**2、子弹类（ClassBullet）**

子弹类公有继承于单元类，继承的成员变量及函数不再赘述。

**成员变量：**

①发射来源的坦克编号k，类型为int；

②是否已经爆炸HasExploded，类型为bool；

**成员函数：**

①子弹构造函数；

②子弹赋值构造函数；

③子弹爆炸函数BulletBomb——用于绘制爆炸效果；

④子弹移动函数BulletMove；

⑤发射来源编号设定函数；

⑥发射来源编号返回函数；

⑦是否已经爆炸设定函数；

⑧是否已经爆炸返回函数；

**部分成员函数实现：**

①子弹爆炸函数：

将已经爆炸的子弹绘图坐标、方向、贴图编号下标、为大爆炸或小爆炸等信息存

入绘图类类的爆炸效果容器中，为绘制爆炸效果做准备。

②子弹移动函数：

首先进行坐标对齐，将坐标四舍五入修正为正确的行和列。由于在调用判断子弹移动之前已经判别了子弹和坦克、子弹和子弹的碰撞，因此只需要判断原始地形的移动是否可行。如果可行则修改对应坐标。

**3、坦克类（ClassTankBasic）**

子弹类公有继承于单元类，继承的成员变量及函数不再赘述。

**成员变量：**

①装甲等级armor，类型为Armor；

②生命值HP，类型为int；

③剩余可以发射的子弹数量RemainingBullets，类型为int；

④是否处于运动状态IsMoving，类型为bool；

⑤绘制的坦克图片编号ShowPictureNum，类型为bool；

⑥坦克编号TankID，类型为int；

⑦暂停运动时间sleeptime，类型为int；

⑧道具计时HaveProp，类型为int（用于决定绘制的颜色，达到闪烁效果）；

⑨保护罩持续时间UnconquerableTime，类型为int；

⑩发射炮弹后剩余冷却时间CoolingTime，类型为int；

**成员函数：**

①构造函数；

②发射炮弹函数Fire；

③移动函数Move；

④装甲等级设定函数；

⑤装甲等级返回函数；

⑥生命值设定函数；

⑦生命值返回函数；

⑧剩余可以发射的子弹数量设定函数

⑨剩余可以发射的子弹数量返回函数；

⑩是否处于运动状态设定函数；

是否处于运动状态返回函数；

绘制的坦克图片编号设定函数——当坦克移动时切换图片，产生履带效果；

绘制的坦克图片编号返回函数；

坦克编号设定函数；

坦克编号返回函数；

暂停运动时间设定函数；

暂停运动时间返回函数；

道具计时设定函数；

道具计时返回函数；

保护罩持续时间设定函数；

保护罩持续时间返回函数；

发射炮弹后剩余冷却时间设定函数——发射炮弹后有一定时间不能发射；

发射炮弹后剩余冷却时间返回函数；

**部分成员函数实现：**

①发射炮弹函数：

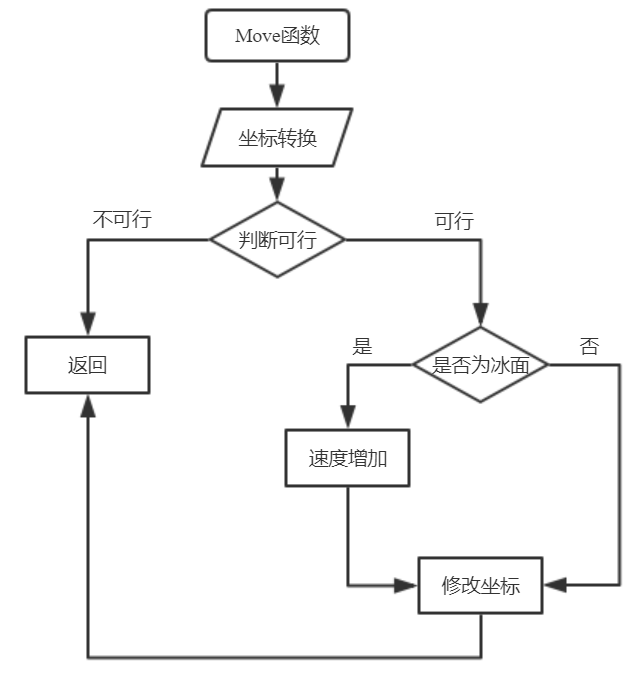
坦克发射炮弹时，必须要满足可以运动且冷却时间已经为零两个条件。

如果坦克装甲等级为LIGHT，那么子弹速度为HighSpeed，否则为普通速度BulletSpeed；将新发射的子弹加入到绘图类的子弹容器中，并重置冷却时间和剩余子弹数量。

②移动函数

首先调用坐标转换函数将绘图坐标转换为地图坐标。

接下来进行坐标对齐，将坐标四舍五入修正为正确的行和列。两次调用移动判断函数，判断物体能否成功移动。第一次是判断原始地形（砖墙、防爆门等）移动是否可行，第二次是判断其他地形（坦克等）移动是否可行。如果都可行则修改对应坐标。需要注意的是如果坦克当前所在单位位于冰面，那么速度增加。



**4、道具类（ClassGameProps）**

**成员变量：**

①绘图横纵坐标x、y，类型为int；

②道具种类Type，类型为int——0代表star效果，1代表shovel效果，2代表clock效果，3代表bomb效果，4代表hat效果，5代表tank效果；

③是否可见Visibal，类型为bool；

④子弹道具出现及消失时间CoolingTime，类型为int——用以表现闪烁；

⑤是否消失Disappear，类型为bool；

**成员函数：**

①构造函数；

②虚析构函数；

③绘图坐标设定函数；

④绘图坐标返回函数；

⑤道具种类设定函数；

⑥道具种类返回函数；

⑦是否可见设定函数；

⑧是否可见返回函数；

⑨子弹道具出现及消失时间设定函数；

⑩子弹道具出现及消失时间返回函数；

是否消失设定函数；

是否消失返回函数；

**5、地图类（ClassMap）**

**成员变量：**

①地图信息Map，类型为int [][]；

**成员函数：**

①加载地图函数；

②修改地图函数；

③地图内容返回函数；

④地图首地址返回函数；

**部分成员函数实现：**

①加载地图函数：

文件读入map.txt中的内容并定位到相应关卡，以十六进制方式读入到Map中。

**6、绘图类（ClassPaintPicture）**

**成员变量：**

①绘制所需图片，类型为IMAGE（包括坦克图片、地图元素图片、子弹图片、爆炸图片、道具图片以及相应掩码图等等）；

②玩家坦克保护罩持续时间ShovelTime，类型为int；

③存储地图信息，类型为vector（包括丛林、坦克、子弹、道具和爆炸的位置）；

④统计信息，类型为int [][]（包括玩家当前击杀各类坦克的个数和总得分，用于

计分板展示）；

**成员函数：**

①加载绘制所需图片函数LoadPicture；

②寻找地图丛林位置函数FindJungle；

③绘制地图丛林函数PaintJungle；

④绘制游戏界面（不包括右侧面板）PaintGameInterface；

⑤绘制游戏右侧面板PaintDisplayInterface；

⑥绘制子弹及移动函数BulletShowAndMove；

⑦绘制爆炸效果函数PaintBoom；

⑧绘制坦克PaintTank；

⑨绘制单个坦克PaintaTank

⑩绘制道具函数PaintProps；

绘制游戏计分板函数PaintScoreInterface；

绘制游戏失败结束函数PaintGameOver；

绘制游戏倒计时函数PaintClockDown；

坦克获取道具函数TankAndProps；

子弹和子弹、子弹和坦克的碰撞函数CollisionBetweenBulletsAndTanks；

地图改变函数ChangeMap；

坦克移动开炮策略函数（简单版本）EnemyMoveAndFire；

坦克移动开炮策略函数（困难版本）EnemyMoveAndFireAI；

玩家坦克击中坦克数量设定函数；

玩家坦克击中坦克数量返回函数；

玩家坦克击中坦克数量清空函数；

玩家积累得分设定函数；

玩家积累得分返回函数；

玩家坦克保护罩持续时间设定函数；

玩家坦克保护罩持续时间返回函数；

**其他数据类型：**

①结构体Boom：存储爆炸效果绘图坐标、图片下标、方向和是否为大爆炸等信息。

**其他函数：**

①判断是否击中函数BeHit：

根据方向判断子弹是否打中坦克。

②绘制字符串、数字函数PaintString：

对于字符串，根据初始位置依次输出；

对于数字，选择输出方式（从左向右或是从右往左）依次输出。

**部分成员函数实现：**

①绘制地图丛林函数：

由于丛林始终在整个游戏界面的最上层，且固定不会发生改变。因此我们只需要存储一次即可。每次地图更新最后调用绘制地图丛林函数，在最上层绘制丛林效果。

②绘制游戏界面：

依次遍历地图的每一个点，取出该处的值输出相应图片。如果为砖块则需要判断是否残缺，若残缺则擦除相应的块。（这里可以加一张图片讲砖块）另外判断总部的情况，如果四个最小块的值之和不为60，则说明有至少一块已经被打掉，此时总部已经被击中，因此绘制被击中的图片。反之输出完整的图片。

③绘制游戏右侧副面板：

首先绘制敌人坦克的缩略图。按一定坐标输出剩余敌人个数的小图标。再分单人和双人模式绘制玩家生命等内容。最后绘制关卡数等信息。

④绘制子弹及移动函数：

由于坦克绘图坐标为坦克左上角的坐标，因此首先要将子弹绘图坐标做一定修正使之位于坦克炮口处后绘制。接下来检查目前在容器的所有子弹，对每个子弹调用两次移动判断函数。第一次是判断原始地形（砖墙、防爆门等）移动是否可行，第二次是判断其他地形（坦克等）移动是否可行。如果均可行则移动并保存；否则说明子弹将遇到障碍物爆炸，即如果第一次判断不可行，说明打中建筑，需要改变原始地形并触发小爆炸；否则说明打中坦克，需要改变其他地形并除法大型爆炸。

⑤绘制爆炸效果函数：

对于容器中的每一个爆炸效果，根据方向适当调整爆炸效果的中心点，修改下一时刻的爆炸效果图片的下标。如果已经到了爆炸效果的尽头则不再存储该效果。

⑥绘制坦克函数：

先对每一个坦克可能的保护罩、暂停运动和子弹冷却时间进行修改。接下来对每一个存活的坦克进行判断，如果是新生成的敌方坦克则贴星星图片，如果是新生成或是处于保护罩状态的玩家坦克就覆盖人物图片，另外如果坦克携带道具就特殊贴图处理；其他情况就正常按照相关信息绘制坦克。对于每一个不存活的坦克如果携带道具则将道具放入容器中。将存活的坦克保存在容器中。

⑦绘制游戏计分板函数：

对本关玩家获得得分先进行累加，然后对具体数据进行动态展示。具体表现为每闪烁一次得分相加该类坦克的单个得分，当全部坦克的对应分数都不再修改后结束闪烁。输出统计的总个数。

⑧坦克获取道具函数：

对每一个存活的玩家坦克和每一个道具容器进行判断，先将坦克坐标转换为地图坐标，暴力判断如果坦克的某一个最小块与道具重合就视为获取到道具，获取以后触发道具特效，设置该道具消失。循环全部结束以后将没有消失的道具保存。

⑨子弹和子弹、子弹和坦克的碰撞函数

首先判断子弹和子弹的碰撞：暴力判断两子弹的横纵坐标关系，当两子弹有一相等且方向不相同时，如果两者距离相差较小时，视为碰撞并引发小爆炸。

子弹和坦克的碰撞思想类似：暴力判断两者的关系，坦克在处理前先将坐标转化为地图坐标，通过调用是否打中函数进行判断，如果打中且子弹不是自己发出的，那么分类讨论：

（a）坦克为玩家，子弹来源于敌人：引发大爆炸效果。此时玩家如果不在保护罩状态，那么减血并回到起始点；

（b）坦克为玩家，子弹来源于玩家：引发大爆炸效果。设定玩家变为初始装甲等级且一定时间不能移动。

（c）坦克为敌人，子弹来源于玩家：引发大爆炸效果。此时敌人如果不在保护罩状态，那么减血。如果此时敌人血量为0，则记录玩家击杀数。

⑩地图改变函数：

同移动判断函数选择两个坐标点，取地图上两点的坐标值。

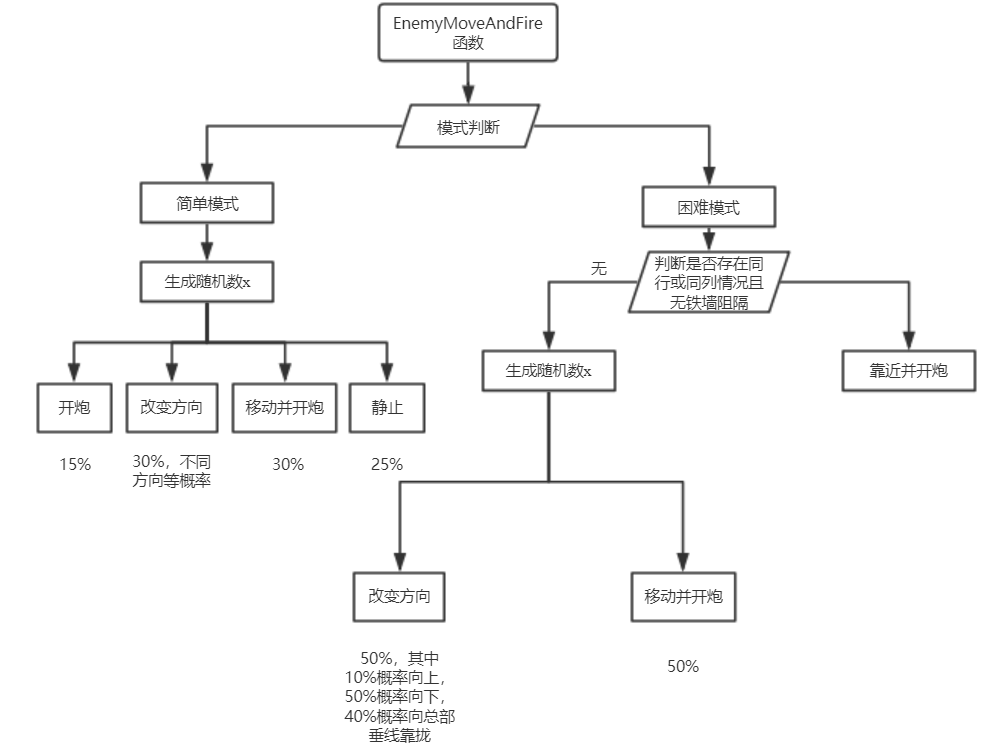
如果有一个是边界，直接返回。

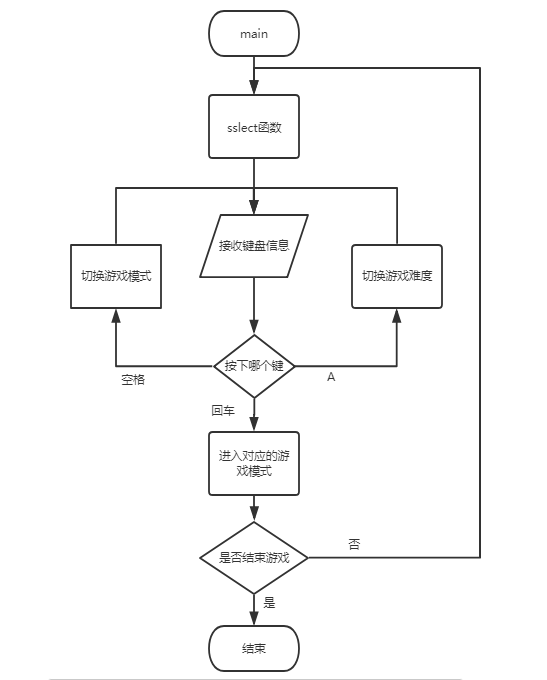
如果打中总部，那么该处置零。

遍历寻找该子弹的来源以确定子弹威力，如果是重坦克且打到的是砖墙，直接置零。否则只能打砖墙。按照击中部分修改砖墙的形状。

坦克移动开炮策略函数

对于简单版本来说，移动开炮策略是随机的。而对于困难版本来说，如果坦克与玩家坦克之一或者总部位于同一行或列时，如果坦克为重坦克直接朝向移动并开炮；如果不是重坦克且中间没有铁墙阻隔也是同理。如果不满足上述条件那么按照随机处理。随机化处理使用了一些技巧，为了避免玩家移动坦克做出相应的策略而产生的反复，因此我们选择随机函数是有一定针对性的。坦克移动和转向概率各占一半，而转向中只有10%向上移动，50%向下移动，剩余按照竖直方向靠近总部的方向移动。这样做有一定的道理，因为只要与总部在同一行或者同一列就会触发AI策略，因此我们这样设定占比。



****

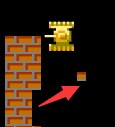
主函数流程示意图

**三、运行结果及分析**

下面依次测试游戏界面和各个模式。（使用32位WIN7系统）

**1、地形测试:**

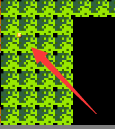
对于原版坦克大战的各种地形都成功实现。



砖块：实现了砖块的部分破坏，一炮只能打碎一层。



铁块：只有三星坦克可以破坏。



草丛：坦克在下面走的时候很难被发现，但是可以分辨。



水面：子弹可以飞过，坦克不可通过。



冰块：坦克在冰上走的时候移动很快。

**2、坦克形态测试：**

**敌方坦克：**

基础坦克高速坦克高速炮弹坦克装甲坦克

高速坦克的移速很快，高速炮弹坦克子弹速度很快，装甲坦克有三条命，每条命有不同的皮肤。装甲坦克的皮肤如下：

**己方坦克：**





**3、道具测试：**

atank

坦克可以给玩家加一条命。

bomb

炸弹可以消灭地图上的所有敌人。

Shovel

铲子可以给老窝一圈的砖变成铁块一段时间。

star

星星可以给玩家坦克升级。一星坦克子弹加速，二星坦克可以连发，三星坦克可以打动铁块。

hat

头盔可以给玩家坦克加上无敌保护壳，一段时间内不会受到伤害。

样式如下：

ProtectiveCover1ProtectiveCover2

以上道具均测试完毕，游戏中顺利实现各种道具的功能。

整体效果如下：



**4、游戏过程测试**

**1P简单模式测试：**

左下角是操作说明，右下角是模式选择模块，一共有四个模式。

右下卡通人物的小花和心表示坦克大战模式敌人难度是简单。

使用学院IOT中的’O’表示当前模式是1P坦克大战。



按下回车进入1P简单模式的坦克大战：

测试结果如下：

经反复核对地图的破坏，道具的闪烁，子弹对坦克的伤害，坦克和子弹的移动，出生坦克的星星特效，道具坦克的闪烁效果并且携带江大校徽均成功实现。



游戏界面1



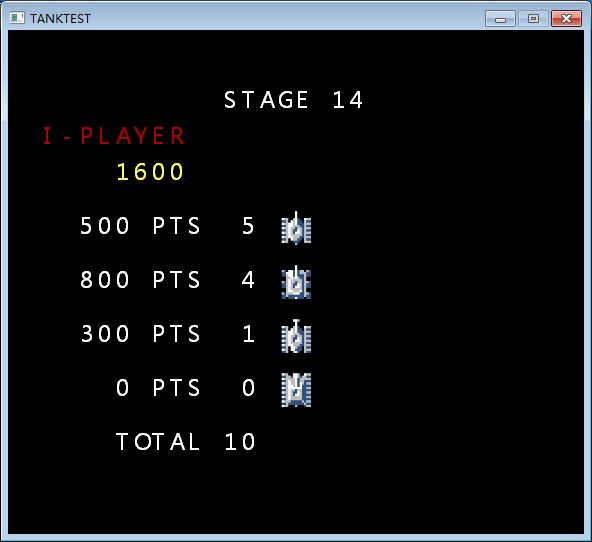
游戏胜利界面2



游戏界面3：



游戏界面4：



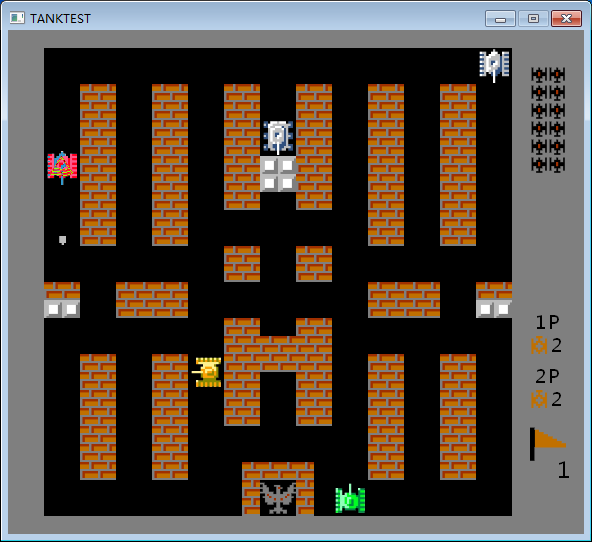
游戏统计界面

**2P困难模式：**

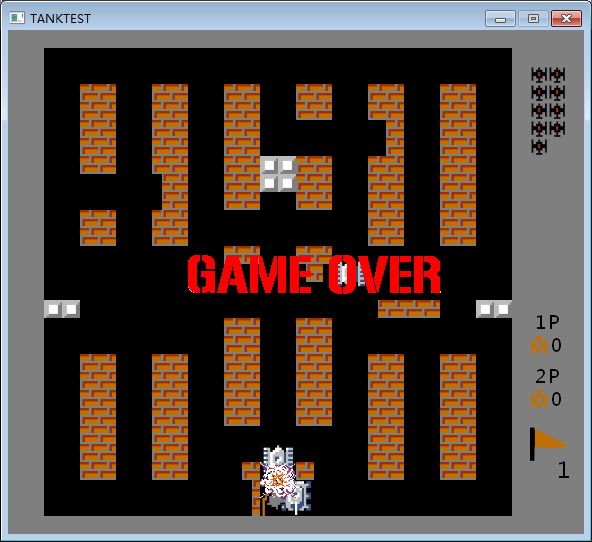
按空格切换关卡，O位置改变。按A切换到高难度，卡通人物拿起武器，表示敌人的难度大大增加。



进入困难模式



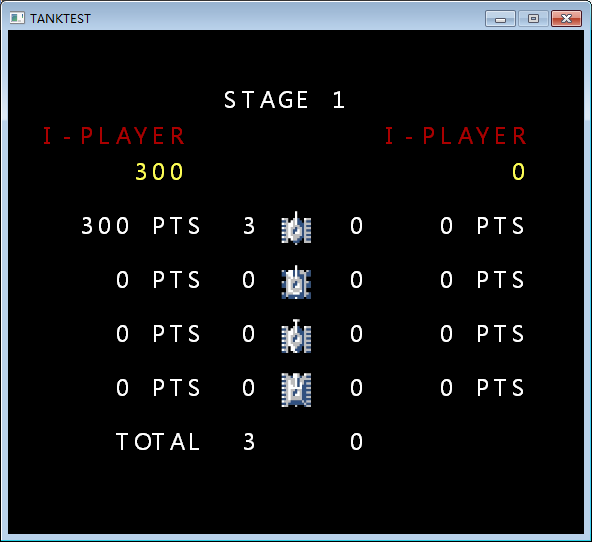
经反复核对地图的破坏，道具的闪烁，子弹对坦克的伤害，坦克和子弹的移动，出生坦克的星星特效，道具坦克的闪烁效果并且携带江大校徽均成功实现。



游戏失败截图



游戏胜利截图



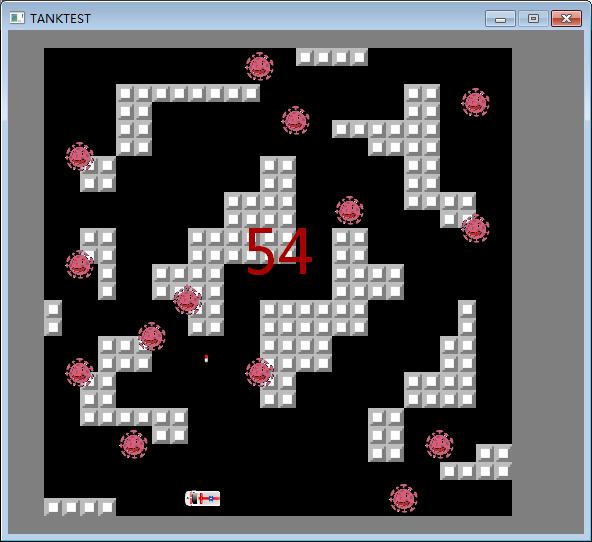
计分板

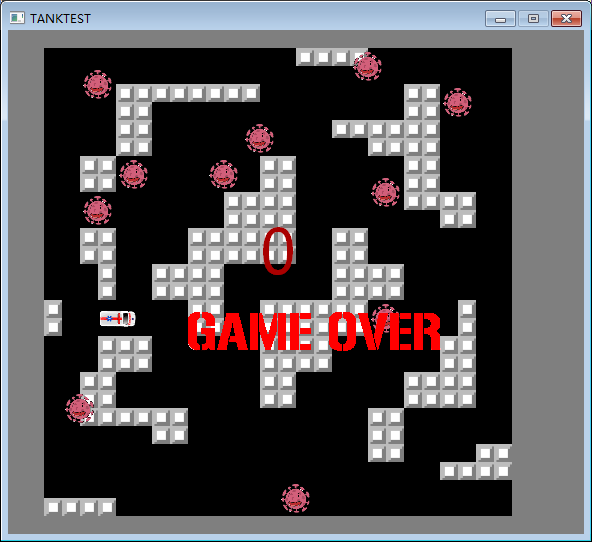
**趣味模式测试**



玩家要控制救护车发射胶囊炮弹，在限定时间内清理所有病毒。

病毒的移动和炮弹胶囊的打击效果测试结果正常





如果没有全部清除，时间截止后游戏失败



游戏胜利



选择模式



游戏开始



游戏过程：病毒进入下方后，幸存者会减少

玩家可以通过发射胶囊或者埋炸药包的方式消灭病毒，此模式为娱乐模式，抗疫必胜。



游戏胜利

**效率分析**

分别对经典模式，CLEARANCE模式，DEFENCE模式进行效率分析。

**经典模式:**

时间复杂度:

每次循环间隔十毫秒，每次循环内依次执行:

地图上坦克不足时添加新坦克——绘制底板——玩家移动和开炮——敌人移动和开炮——坦克地图的更新——子弹和子弹以及子弹和坦克的碰撞——子弹移动——坦克绘制——爆炸效果绘制，丛林绘制——坦克获取道具——绘制道具——游戏结束检查（如果满足则加载下一关）。

时间复杂度为O(1):添加新坦克、绘制底板、玩家移动和开炮、坦克地图的更新。

时间复杂度为O(n):敌人移动和开炮、子弹移动、坦克绘制、爆炸效果绘制、丛林绘制。

时间复杂度为O(n2):坦克捡道具、子弹和子弹以及子弹和坦克的碰撞。

因为数据规模很小，几乎所有的n都是一位数，所以在时间复杂度上，我们的代码可以跑得很快，提供非常流畅的游戏体验。

空间复杂度：

我们需要把存储图片信息的IMAGE 数组。存放坦克信息的VTank变长数组，存放爆炸未知的VBomb变长数组,存放子弹信息的VBullet变长数组。

空间复杂度为O(1)的有：IMAGE 数组。

空间复杂度为O(n)的有：存放坦克信息的VTank变长数组、存放爆炸未知的VBomb变长数组、存放子弹信息的VBullet变长数组。

数据规模非常小，我们的算法可以顺利实现。

**CLEARANCE模式**

时间复杂度：

时间复杂度为O(1): 绘制底板、玩家移动。

时间复杂度为O(n): 病毒移动、胶囊移动、救护车和子弹绘制、爆炸效果绘制。

时间复杂度为O(n2): 病毒和胶囊的碰撞。

空间复杂度：

空间复杂度为O(1)的有：IMAGE 数组。

空间复杂度为O(n)的有：存放病毒的VTank变长数组、存放爆炸未知的VBomb变长数组、存放胶囊信息的VBullet变长数组。

**DEFENCE模式**

时间复杂度：

时间复杂度为O(1):添加新病毒、绘制底板、玩家移动。

时间复杂度为O(n):敌人移动、胶囊移动、坦克绘制、爆炸效果绘制、胶囊绘制。

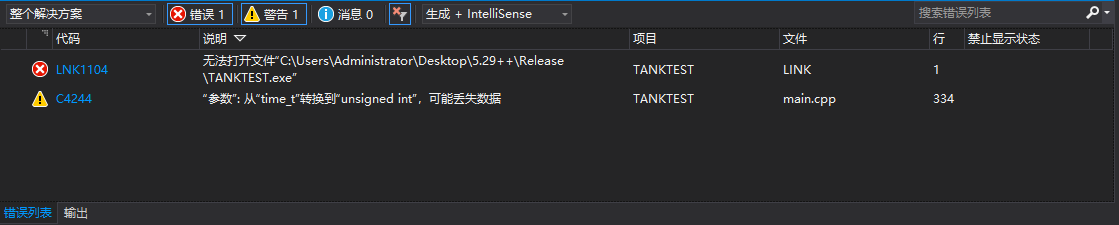
时间复杂度为O(n2):胶囊和病毒之间的碰撞、药箱和病毒之间的碰撞。

空间复杂度：

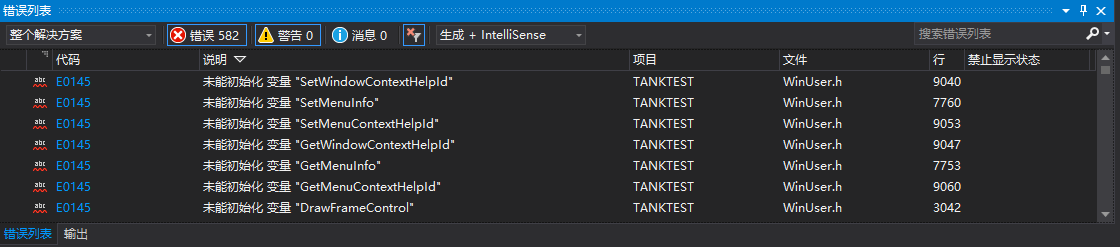
空间复杂度为O(n)的有：存放病毒信息的VTank变长数组、存放爆炸未知的VBomb变长数组、存放药箱信息和胶囊信息的VBullet变长数组。

**四、调试过程**

1、代码已经运行之后又重新点击运行。

****

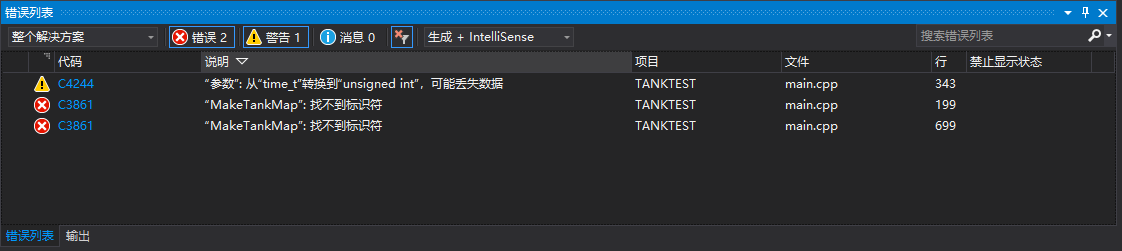
2、没有include库函数graphics.h。



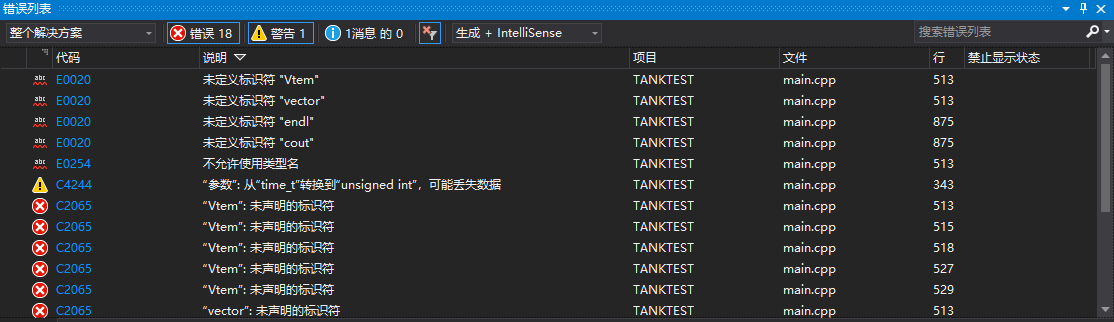
3、没有对函数进行重载，导致传参个数报错。



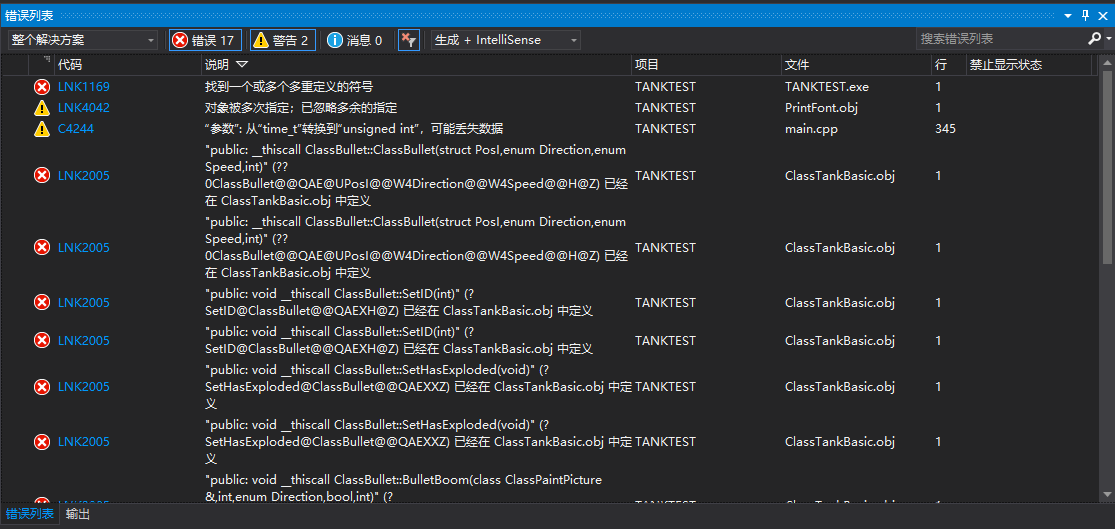
4、函数在使用之前没有声明报错了。



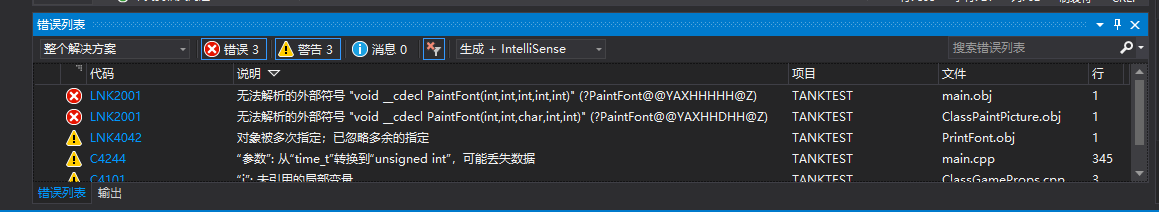
5、没有使用标准命名空间using namespace std;



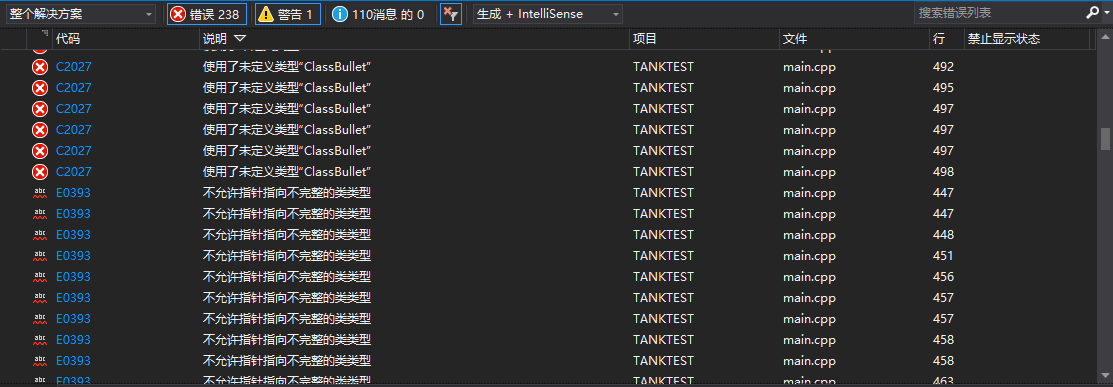
6、.h文件和.cpp文件在include时出错。要尽量避免在.h中导入头文件，可以用前向声明代替。



7、每次第一次编译代码都会出现这种错误，可以点开相应的.h和.cpp文件，ctrl+s保存之后，第二次编译运行就可以了。



8、没有include相应的头文件 或者是没有前向声明这个类引起报错。

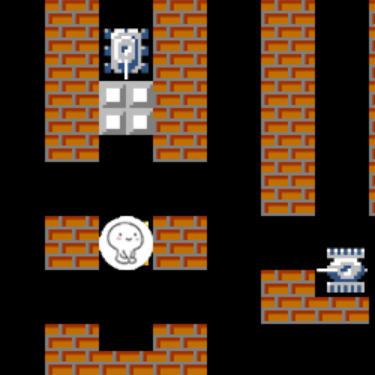
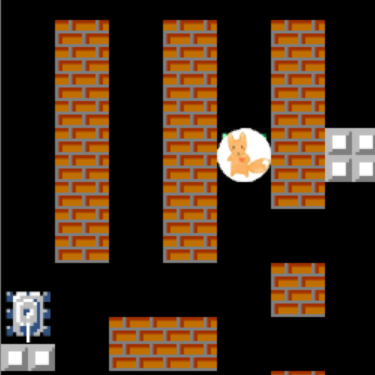


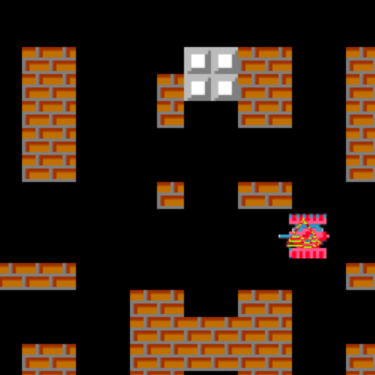
**五、突出亮点**

**1、游戏动画的创新：**

游戏动画是我们程序设计的一大创新点。我们在借鉴原版FC坦克大战的基础上充分发挥想象力，加入了许多我们自己的元素。例如玩家的保护罩设计为我们的卡通头像、带有道具的敌人会有江南大学的校徽等等。

（这四张图我会想办法放在一页）

**2、游戏模式的创新：**

我们设计了两个打病毒的趣味模式，既有一定的趣味性，又与当下时代热点契合，也表达出我们对于抗疫胜利早日到来的殷切期盼。在趣味模式中，玩家化身为救护车向病毒发射药丸、安置医药包，击退病毒。

CLEARANCE MODE中，玩家需要在60秒内消灭地图上全部共计15个病毒，病毒每隔一段时间会随机移动。玩家需要有一定的反应能力和投弹技巧。

DEFENCE MODE中，玩家需要保护在安全线后面的“我们”，击退汹涌而至的病毒。这个趣味模式没有失败的局面，意味着这场战疫我们必将取得胜利。

在这两个模式中的元素都具有创新性，或者是同学的头像，或者是全新的背景和道具图片，能够吸引玩家的兴趣。

**3、游戏界面的创新**

相较于原版坦克大战，我们对于游戏界面也做出了一些调整。一方面，在游戏模式的选择界面加入按钮以及对应的操作示意，使得玩家能够更好的上手，提升用户体验感。另一方面针对于不同难度的经典模式，玩家所看到的卡通人物也有变化。这样的呈现比简单的“Easy”和“Hard”的提示更加生动和有趣。至于原版坦克大战模式中模式选择的指示坦克，我们将其修改为物联网工程学院的院徽一部分。既体现了游戏的趣味性，又体现了设计的特色。

**六、心得体会**

杜昊：最大的收获是有了一次和其他人组队做项目的经验。相比于传统的做题还是有很多区别的。在完成这份作业的过程中，我时常在寻找资料的时候学到了很多，也从队友身上学习到不少经验。在完成最基础的功能以后，我们就在思考如何改进算法，增加新的功能。因为做题看到Accepted之后就万事大吉了，而一个项目往往没有好坏之分，需要反复的修改和改进。

孙创昱：这次和队长一起写大作业的过程中，巩固了课堂上所学的知识，对语法的记忆更加清晰了，对C++的特点也有了更多的了解。本次比较遗憾的是没有用到老师之前介绍过的GitHub，而是采用QQ群的方式保存代码，以后尽量多用GitHub，要经常保存代码，否则如果写错了，就有可能重新开始。