大作业报告

2018200222 查王皓天

**总体概述**

在经历了长时间改bug，调数据，优化后，终于完成了这份大作业。基本达成老师的所有要求，完成了所有的基础功能和附加功能，以及一些自己心血来潮加的小功能，并且贯彻了OO的思路（确实可以非常好的实现代码的重用），运用了QT的窗体类、控件、槽函数以及尝试了一下qt的ui（这个真的很好用，简化很多代码）。虽说画面称不上精美，但也清晰流畅，背景音乐，事件音效、游戏介绍也相应的有做。

游戏方面一共做了三关，每关都是三波敌人，有蘑菇、飞龙、牛头怪三种敌人分别有不同特性，有普通炮塔、狙击塔、减速塔、眩晕塔四种类型的塔，塔可以升级拆除、显示攻击范围、智能索敌，敌人会从两个路径出现。

下面我来详细介绍一下这个游戏的功能、设计和实现。

**项目设计&类设计**

本工程一共有21个类，分别是

按钮类：mybutton,buildtowerbutton

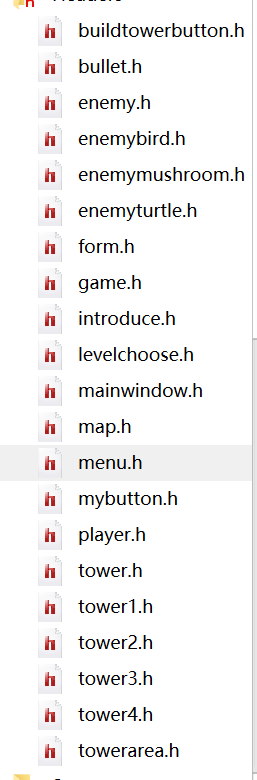
界面类：mainwindow, levelchoose, introduce, form, menu

敌人类：enemy, enemymushroom, enemybird, enemyturtle

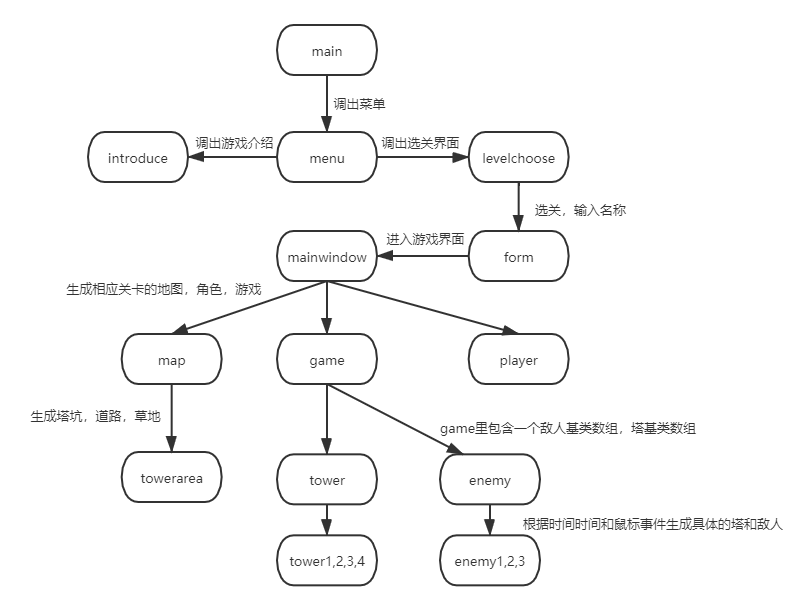
塔类： tower, tower1, tower2, tower3, tower4

子弹类：bullet

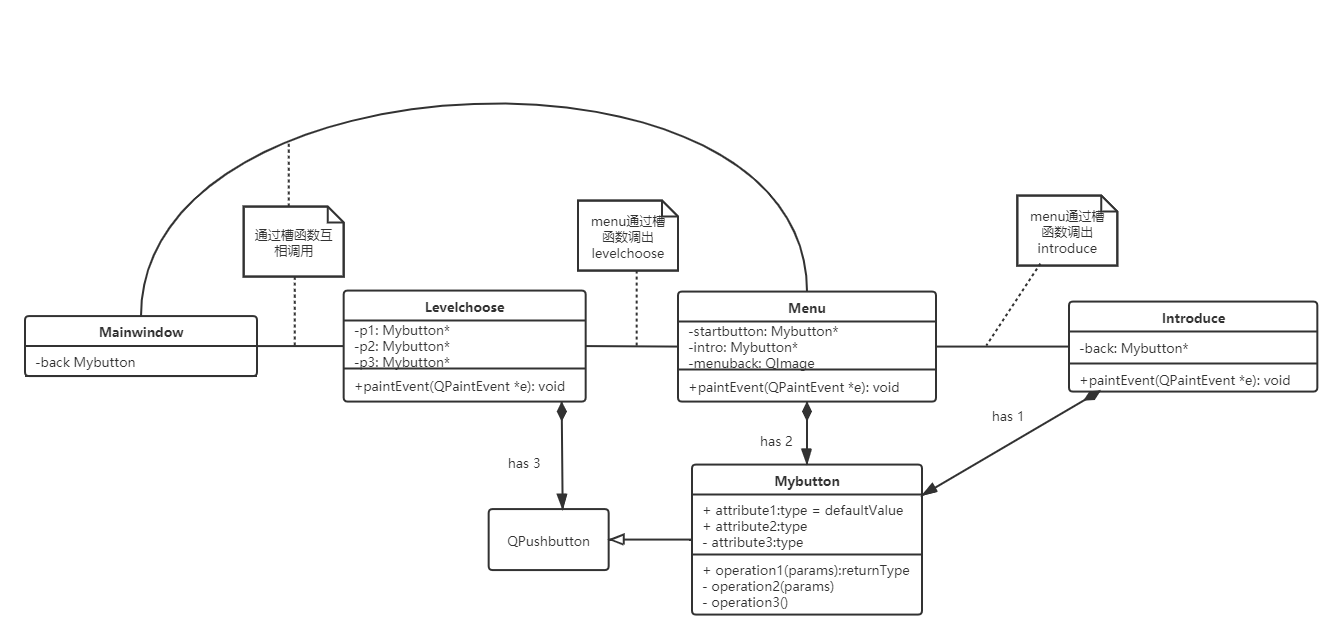
游戏类：game，玩家类：player，地图类：map，main文件



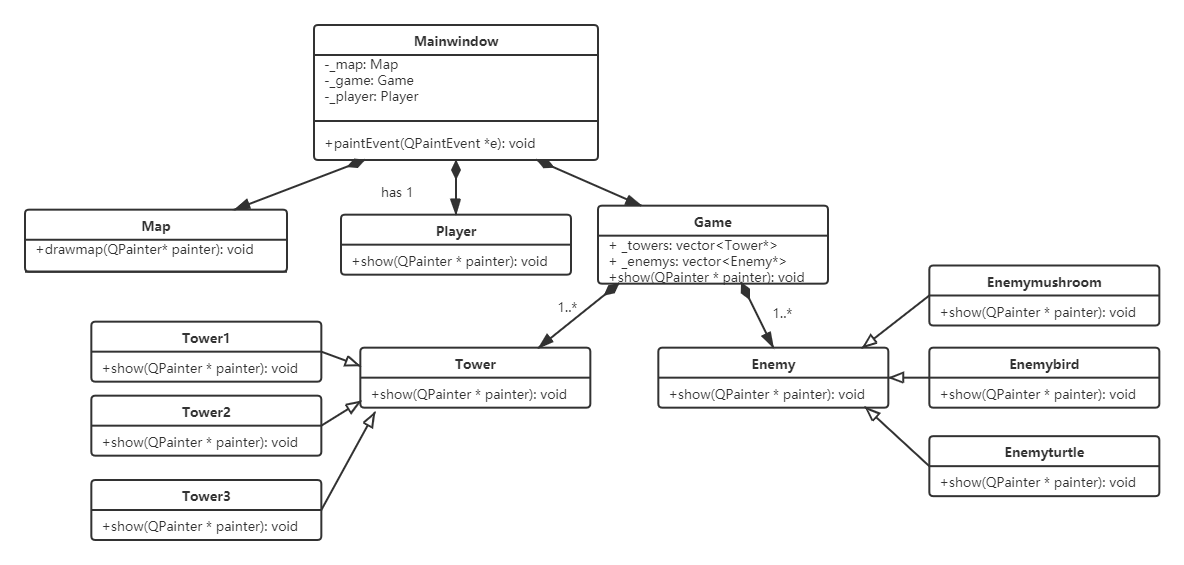
类之间的简单的关系如下图：

下面我分成不同个part用uml图来表现类与类之间的关系与整体设计，这部分主要讲的是程序的设计，具体如何实现会在第三部分来讲。

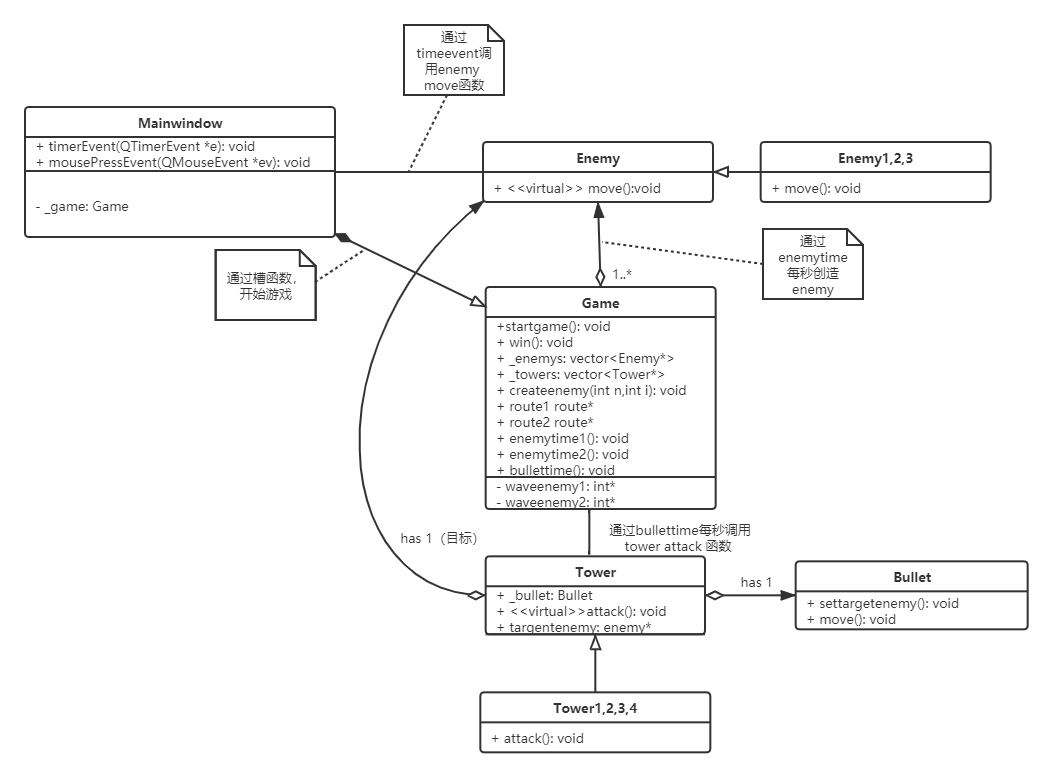
界面部分

上图是界面部分的uml图，省略了窗体之外的函数和类成员，一切都是从menu窗体开始，然后通过qpushbutton和mybutton以及相关的槽函数调用游戏介绍界面、选关界面和游戏界面等等。

画图部分



画图部分的uml图关系如上

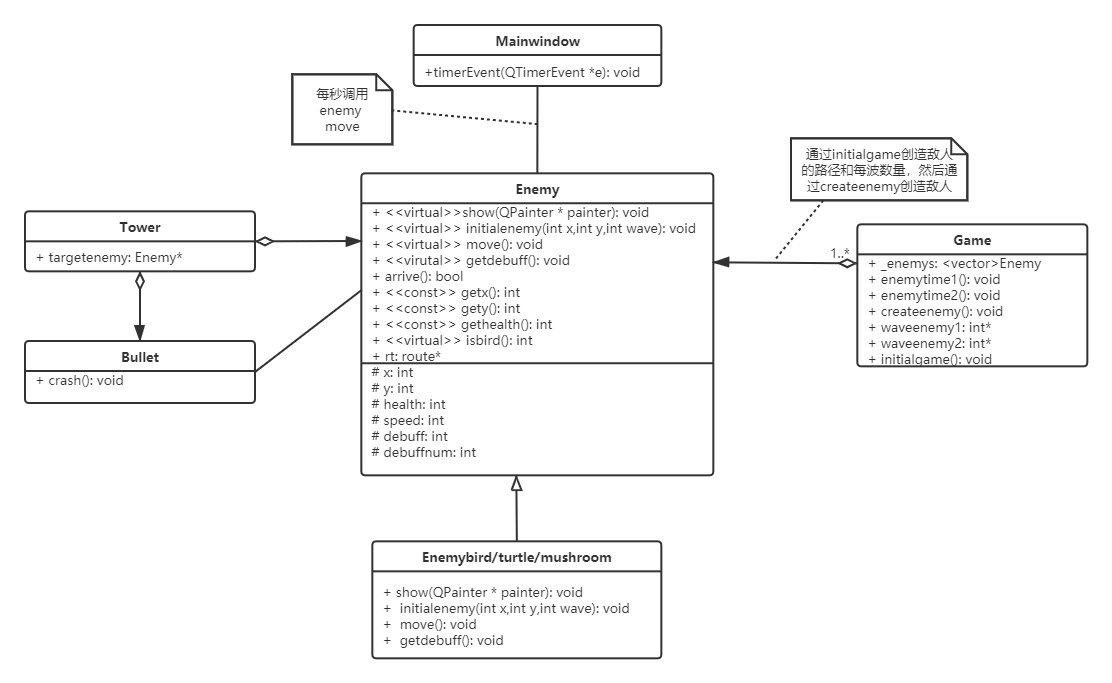
游戏流程部分

这部分是整个项目的主体，关乎游戏是怎么运作的，首先当我们完成上述的画图部分，就得到了静态的内容，然后，通过点击控件就开始了一局的游戏。

这时候一共有4个计时器会启动，分别是enemytime1,enemytime2对两路创建敌人，timeevent控制敌人按照预设的路线移动，然后根据敌人的血量、位置决定是否消除敌人，bullettime会让塔搜索敌人，并发射子弹攻击

这四个计时器使得整个游戏可以动起来，当消灭所有敌人后，游戏获胜。

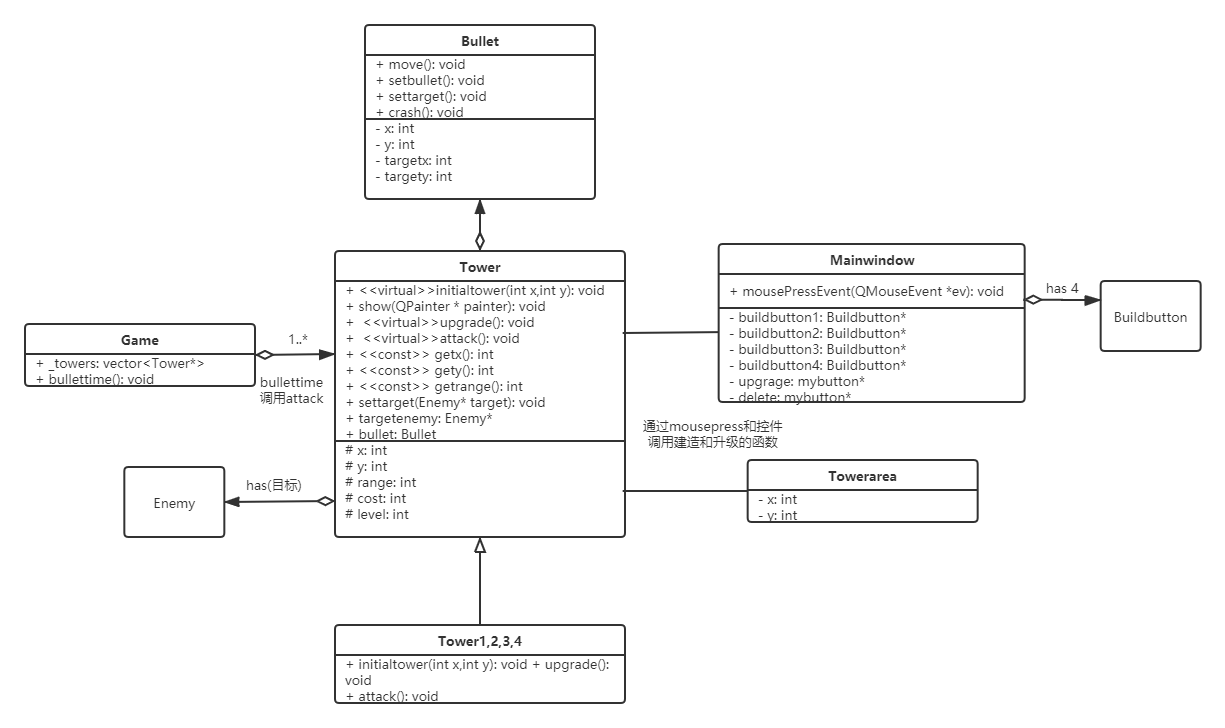
敌人部分

敌人部分Uml图如上，这个部分核心思想是运用C++的继承多态实现代码的重用，关于如何重用后面还会在提到，这里主要讲一下敌人部分的设计。敌人首先是在游戏开始的时候，就会创造两条路径的的敌人波次，分别存放到waveenemy1和waveenemy2两个数组中，同时也会创建两条路径，然后根据定时器和波次数组每秒创建敌人，赋予它路径数组，另外的定时器每秒会调用move函数按照路径移动，然后敌人被塔打就会扣血，如果血量小于零或者到达目的地，敌人就会erase掉。

一共有三种不同的敌人，他们的区别在于血量、速度不同，移动方式不同。如果是有翅膀的敌人，不会根据路径移动，而是直捣黄龙，直奔目的地。

当敌人波次数组改变后，波次会相应的增加，同时敌人被创建的时候会强化。敌人死亡后获得金钱

塔部分



塔的uml图如上，关于塔的设计：首先是塔的创造，我开始想了几种方式：一种是拖拽式，一种是点击式，后来采用了第二种，也就是点击相应位置在那里创建一个塔。然后我引入了塔坑，有以下好处：1.塔坑可以防止塔重合，避免出Bug. 2.塔坑可以更清楚地确定一个地方是有塔还是无塔，以便于调用build按钮还是upgrade按钮。3.塔坑也能增加游戏的战略性。一个塔坑如果没有建塔，就会调用Build按钮，建了塔就调用upgrade和delete按钮。每个塔会根据相应的规则设定一个敌人目标，然后会对敌人目标开炮。

然后塔还有4个子类继承。分别是普通塔，狙击塔，减速塔和眩晕塔，他们在花费，设程，威力上都有差别，后两个塔在攻击的时候，还会给敌人上debuff，减少他们的速度。

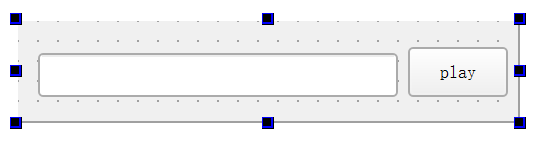
升级塔可以增加射程和威力，拆除塔会返还钱币。

**功能完成情况&实现思路**

菜单界面&&游戏介绍&&选关界面&&输入玩家姓名

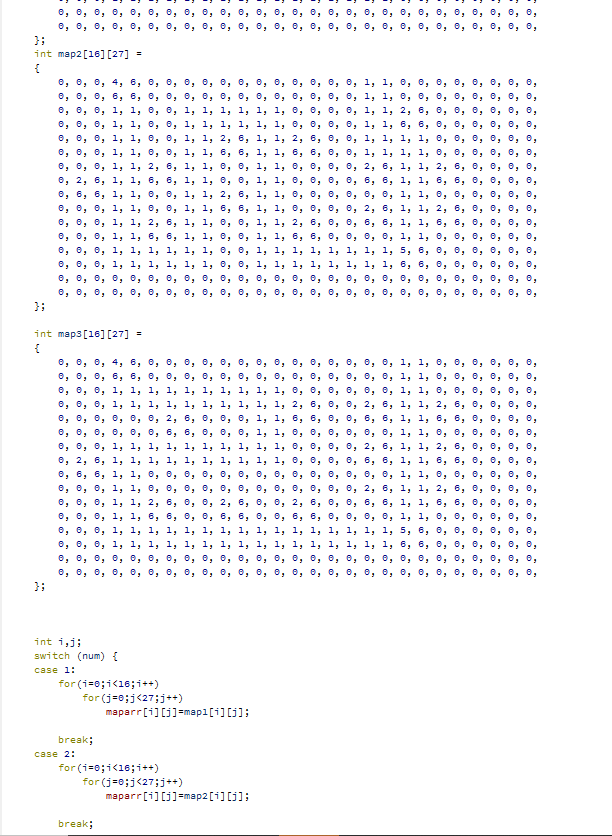
菜单界面比较简单，两个按钮，按下后就会调用关闭此窗口和打开新窗口的槽函数，然后游戏介绍界面我提前做好游戏介绍，将这张图片作为窗口背景，然后有一个返回按钮，可以返回菜单，选关也是3个按钮，会传送关卡参数到后面游戏中，不赘述了。

这里来讲一下输入玩家姓名这部分，这里我尝试了qt的ui功能，直接画了一个lineedit和button，这也是让我比较有感触的地方，就是这个ui非常好用，比我之前手写Button槽函数啥的简单很多，右下角直接调参数。



三个关卡&&不同场景

在我这个简易游戏中，不同的关卡不一样的地方在于：地图绘制不同，敌人顺序、数量不同，敌人路径不同，其他的地方包括敌人种类啊，塔的种类啊都是一样的，所以我创建不同的关卡并不困难，只要把关卡的参数传到上述功能实现的地方，用一个switch case就好了。下面举画地图的例子说明：



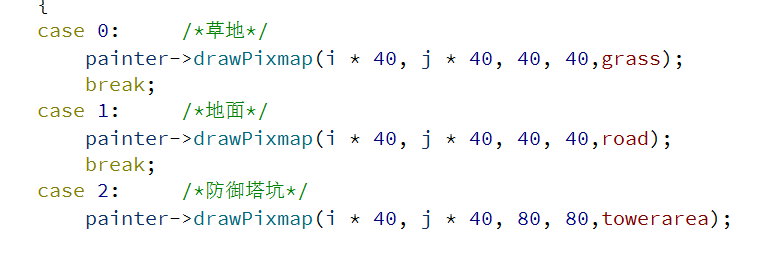
创建地图的时候有三个地图模板，然后根据关卡数对地图数组初始化。

地图格子

如上图，可以看到在我这个游戏中，整个地图是被化成了一格一格的，每一格是40\*40，这个思路是我从网上的代码中获取的灵感，我认为这个设计特别的好。开始的时候我也是用了一张背景图案作地图，后来采取了这种网格式的想法，这样对坐标的定位非常方便，而且坐标都是40的倍数，计算的时候非常精确。

地图绘制

这部分就是根据上面一格一格的思想，事先安排好地图中每一格的内容，然后根据这个内容放一张40\*40的图片进去



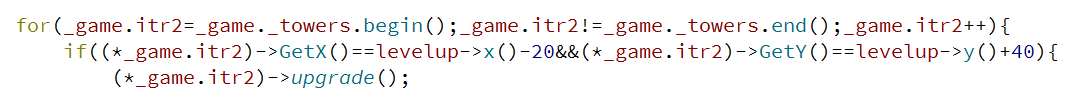
塔的创建/升级/删除

这部分我的思路是有6个按钮（升级/建造/拆除），平时会隐藏起来，需要他们的时候就在相应位置出现。

塔的创建我是采用了点击创建的思想，先遍历塔坑确定鼠标点击位置是否在塔坑上以及在哪个塔坑上，然后遍历一下所有的塔看看有没有和这个塔坑位置重合的，确定该位置上没有塔后，之后就会调出上下左右四个建造按钮，点击创建按钮就建了一个塔。

然后我讲一个小技巧，由于我们根据塔坑位置移动按钮的位置，所以按钮位置也可以反映塔坑的信息，按下按钮就可以在该塔坑建塔。

升级和删除是一样的，如果鼠标点击位置是塔坑且有塔，就会调用升级按钮（无塔是建造按钮），再点击，就会根据按钮的位置反映升级/拆除塔的位置，找到那个塔。



（巧妙地用按钮位置反映塔的位置）

塔的攻击

这部分是我认为本项目有改进空间的地方。

我的想法是，首先每个塔有一个攻击对象（可以为空）。如果对象为空，就会去在攻击范围内寻找一个敌人（我是找到的第一个敌人），如果不为空会做如下判断（一、有没有更好的敌人（弱AI）。二、敌人在不在范围内。经过以上步骤，如果还有敌人对象就会进行攻击。

攻击的实现我是每座塔有一个子弹，每次都会把攻击对象的位置传送到子弹里，然后子弹会根据敌人位置相应的移动，如果击中敌人，敌人扣血，子弹会回到初始位置。

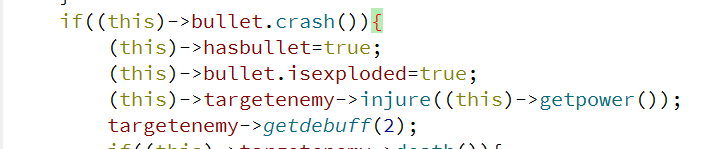
可以改进的地方是：

一、塔的攻击对象改成攻击对象数组，这样可以实现范围攻击型的塔。

二、子弹改成子弹数组，这样可以实现多子弹快射速的塔

塔的debuff效果（减速/眩晕）

这也是不同类型的塔的特点。我的想法是，对于一些塔，改写他们的attack()函数，在击中敌人时，会给敌人增加10层debuff，每次show敌人的时候如果他身上有debuff，相应的数值（速度）会改变，同时也会减少一层（也就是说减速/眩晕的时间是10\*show调用间隔）。

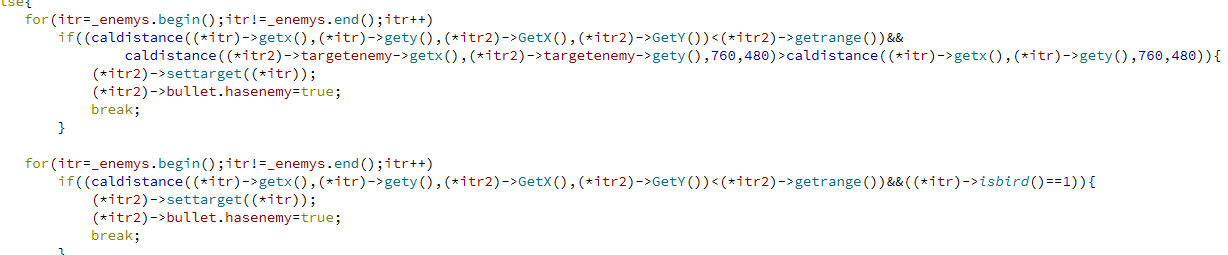


塔的弱AI

我的弱AI实现的是在搜寻敌人方面：

1. 如果有敌人比原本攻击对象离老家更加，会更改攻击对象
2. 如果有飞兵在射程内，会优先攻击飞兵

这部分没什么好说的，直接上代码，就是用一个循环判断



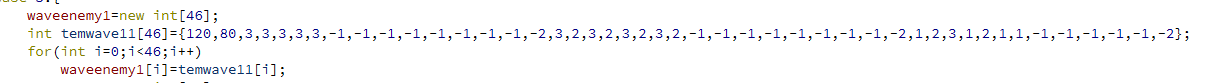
（上面是判断更近，下面是判断飞兵，所以飞兵优先度是最高的）

显示范围

点击控件可以显示范围和隐藏范围，我是game里有一个参数，点击会改变，然后这个参数会决定要不要画塔的时候画范围。这部分我想说的是我发现QSring用双引号中文会有问题，所以上网查了个str2Qstr函数来让中文正常显示。

敌人的生成

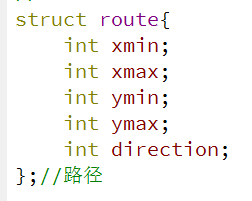
这部分我是用了一个敌人数组加一个计时器，敌人数组在开始的时候会初始化，里面存放了敌人的数量和先后顺序，然后每次调用计时器就会从头搜索第一个非零元，这就是这一次要创建的敌人，创建后把这个位置变成0。



（-1代表跳过，1，2，3是三种敌人）

敌人的移动

在创建敌人的时候会添加路径，一个游戏有两条路，所以一共是两个路径，这个在初始化地时候会搞好，然后路径是一个结构体数组，每个元素是代表当前路径最小最大xy和当前路径的方向，敌人在移动的时候会根据目前位置，判断处于哪一个路径，然后根据方向移动，当移动到下一个路径时，判断的时候就会到下一个路径中去。





（例如左边一条路有6个路径，右边一条路有3个路径）

如果是飞龙的话就会直接朝大本营移动。

敌人的波次

在敌人数组中会有一个-2代表下一波，如果当前位置是-2时就会波数加一，这里有个问题，我判断波数不是用一波敌人被消灭，而是根据时间来，就会出现上一波敌人可能还没死，下一波就开始了（当然这也不能说是Bug吧，根据时间来增加波数也是可以的）

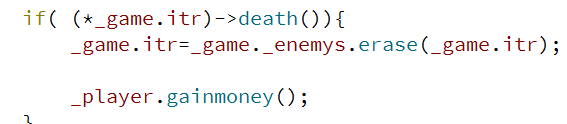
游戏难度随时间增加：

这个实现很简单，创建敌人的时候调用一下当前波次就好了



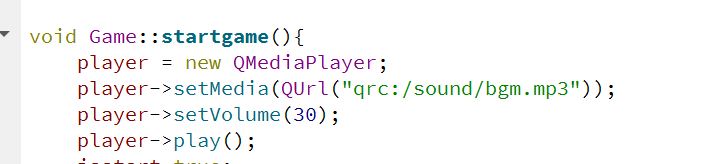
防御方资源的累计：

这个很容易，打死敌人会加钱就好了



背景音乐：

开始游戏的时候回放背景音乐：



事件音效：

我一共是有两个事件，分别是子弹击中敌人会有一个爆炸效果，还有敌人到达目的地会有一个流血效果和音效

前者实现就是子弹如果打到敌人，有一个参数会改变，然后就会触发效果，计时器下一次运行又会把那个参数设回来，所以图片出现的时间就是计时器的间隔

后面那个实现比较简单：



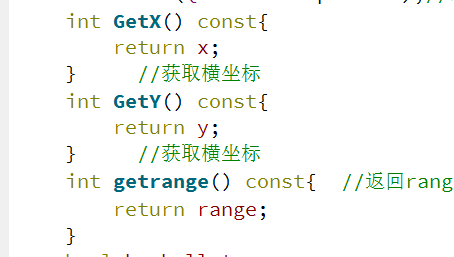
**面向对象思想**

这个作业中我贯彻了OO的思想

封装

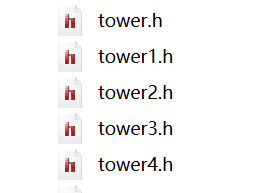
首先是类的封装，在塔、敌人类中我都设置了private参数和相应的接口，安全性高也便于改变。

然后是我将函数放在它合理的地方实现，比如说塔的攻击、敌人生成，并不是在mainwindow里面实现，而是在game里，因为塔和敌人不属于mainwindow，game属于mainwindow，塔和敌人属于game。



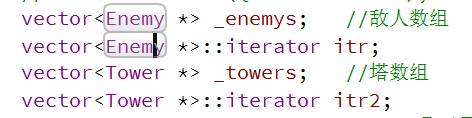
继承

我的继承主要是用于防御塔和敌人，在这两个地方我都做了两个基类：tower和enemy，然后根据具体的要求实现tower1,2,3,4，和enemy1,2,3的继承和改写，大大提高了代码重用率，也实现了不同特点和属性塔和敌人。

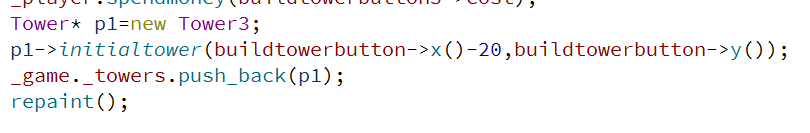


多态

上面的封装和继承还是比较常规的，多态这里我想自吹一波。在这个地方我下了挺久功夫的，最终效果也很好，确实减小了很多代码量。我的塔和敌人都是存放在game文件vector中的，我并没有采用多个vector，而是采用了基类的vector。



所以每次判断都是调用基类的指针，但是在创建对象的时候，用的是子类的类new，基类在很多会变化的地方（move,attack等）采用虚函数的形式。





我认为这就是指针vector和虚函数（多态）的完美结合。首先用一个vector在遍历的时候非常方便，比如我要遍历塔作攻击，我就调用塔vector从头到尾，不管是什么类型的都遍历，然后再调用attack的时候，又因为虚函数和用子类new的原因，就回去调用子类的attack。敌人也是一样，遍历敌人移动的时候是基类指针，但是具体指针调用移动函数又会调用子类的move。真的非常方便。

我开始是这么设想，但是很快遇到了问题，开始的时候用的vector数组下标来调用，但是我发现即使我用的是虚函数和子类new，仍然调用的是基类函数，这真的非常疑惑，后来上网查后改成iterator指示器，居然就好了，可以多态了，具体原因也不大清楚。

**遇到的困难&收获**

在这个大作业的过程中，可以说是遇到了非常多的困难，最绝望的是这些问题并不像以前通过debug就能解决，有的真是莫名其妙（解决的也是机缘巧合），所以我认为完成这样一份大作业真的不容易啊。

问题类型一：代码逻辑的问题

比如我开始写敌人死亡的时候，总是会程序自动结束，我百思不得其解，后来静下心来看代码，发现我删除了指针后，还调用了这个指针，改一下果然就好了

问题类型二：让人抓狂的问题

我在写好程序后，发现如果我的子弹特别多，经常容易死机，子弹少就不会，然后下面提示什么qthread error，我上网查了查也没什么讨论的，我就在想thread是线程，那我的线程太多了？后来我把击中敌人的音效去掉了，就神奇的解决了这个问题，可能是频繁地播放音效导致线程太多，就死机了。这类问题真的很让人抓狂，因为它的知识超出了我的知识范畴，只能通过百度自学。

再比如我换成iterator后，经常打死最后敌人会死机，非常奇怪，后来我上网查才发现原来是用iterator调用erase后，iterator会自动往后移一位，如果最后一个被擦除，就会报错，所以iterator不能写成(for ; ; ++)的形式，而要用if else来++。

我觉得碰到这些超过我知识范围的问题非常棘手，因为我自己一个人捣鼓永远也弄不出来，这时候就需要请教师兄或者老师，或者百度自学。在未来地计算机学习过程中，应该会更频繁地遇到此类问题，所以要多利用身边的CSDN、简书啥的，这也是我的一个体会。

**一些不足&后续可以再改进的地方**

尽管我自己对这份大作业还是比较满意的，还是有一些可以改进的地方，前面也提到了，主要是：

1. 塔的敌人对象可以改成敌人对象数组，实现范围攻击塔
2. 塔的子弹可以改成子弹数组，实现连发攻击
3. 敌人可以用动态图片，实现动态地移动
4. 升级的时候可以改变塔的外观。

Github地址： <https://github.com/zwht81/zwhttowerdefense>

视频地址：https://weibo.com/tv/v/J8MEvnO58?fid=1034:4520935634436105