

高级程序设计：塔防小游戏

程子涵 201220015 project1(phase1)

模仿市场主流塔防游戏，设计了游戏的机制与交互操作。

1.当前进度：

· 实现了主游戏界面、关卡选择界面、主游戏界面及三个界面间的切换。

· 实现了游戏机制：

玩家的胜利是在所有怪物消亡(死亡或走到了终点)后生命值>0。

玩家的失败是在怪物消亡前生命值<=0。

· 实现了一张地图(8×12)：在要求文档中的示例地图基础上，加入了墙和远程塔放置位的设计。

· 实现了敌人移动路径，为可重叠的路径。

· 实现了资源随时间流逝而增加，实现了手动撤回已部署的单位。

· 实现了一个近战塔：地刺。

· 实现了一种敌人：野猪。

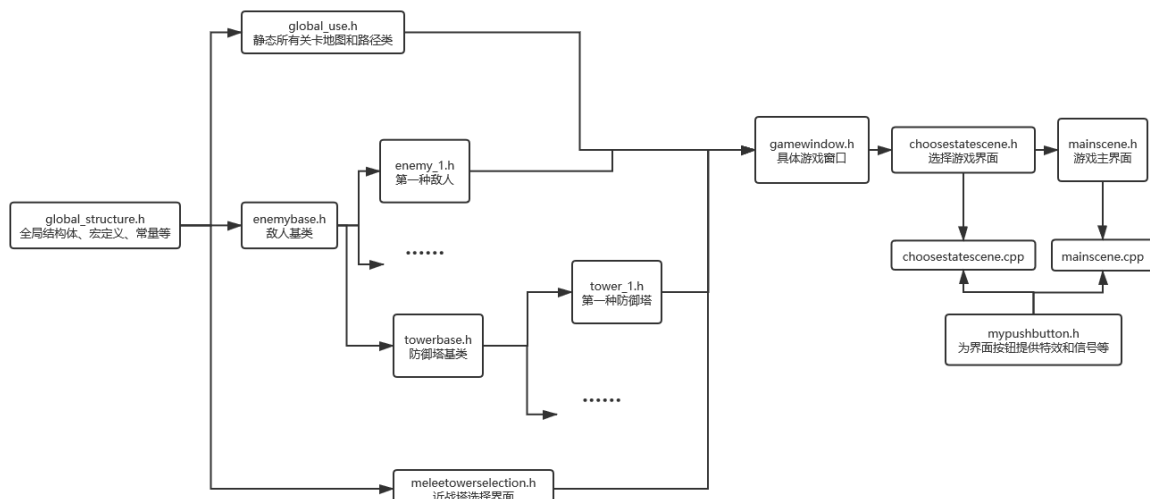
2.整体设计

2.1 项目基本情况：

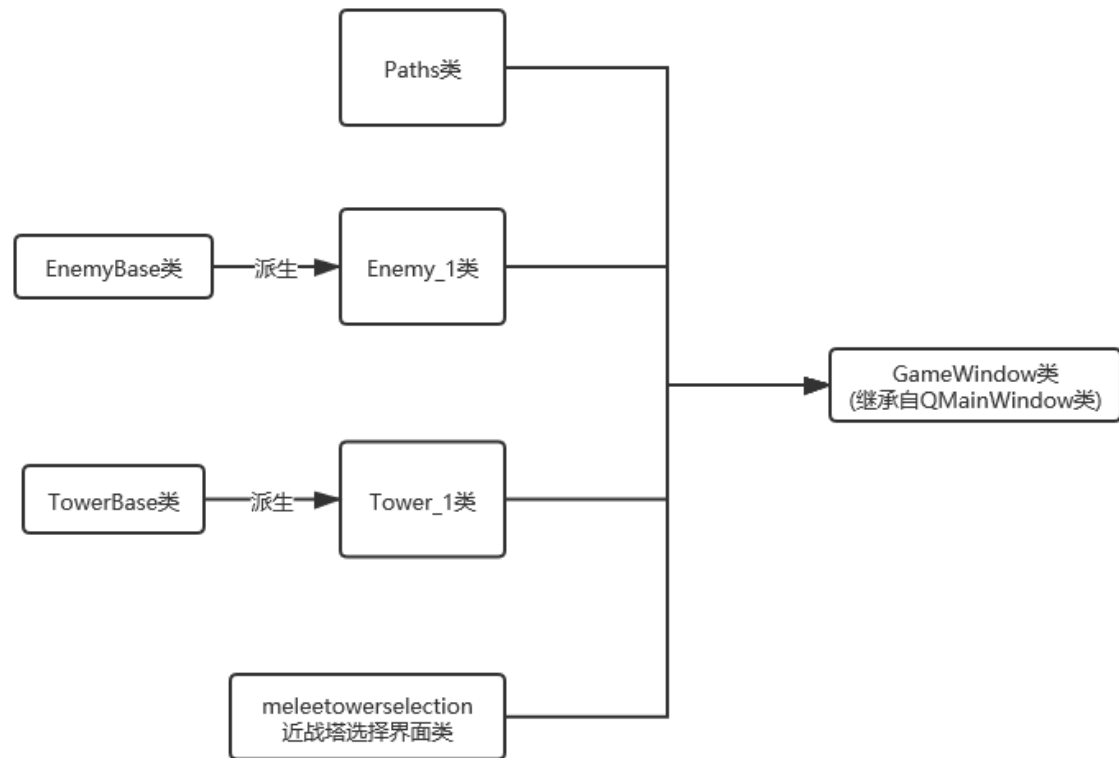
项目使用Qt开发框架完成，IDE为Qt Creator 5.3，使用mingw-win32-gcc482工具集，基于Qt自带的QWidget、QMainWindow等库提供的C++接口完成开发。

2.2项目设计

2.2.1头文件的引用示意图



2.2.2游戏主体使用的类及依赖关系



项目游戏主体使用的类及依赖关系

2.2.3项目设计基本思想（零散）

一个界面就是一个类。

类的逻辑功能执行在构造函数内完成(相当于这个界面的一个main函数)。

逻辑功能完成后,再在类中单独重写paintEvent函数，完成UI图像的显示，和逻辑功能的执行是并行的(图像绘制每次调用update()函数时的游戏图像，可以理解为每次调用update()就是[一帧])。

我把paintEvent理解图像动画显示的总函数，由它来调用每种物件的绘制子函数。

界面间的切换用双向的信号和槽实现(一去一回两个connect):

第一个connect:(父界面的构造函数中就应new好子界面)

信号发送者：父界面类对象

信号接收者：子界面类对象

信号为点击某个按钮clicked

槽为lambda函数，hide父界面再show子界面

第二个connect:(还是在父界面写,监听子界面发出的返回信号)

信号发送者：子界面类对象

信号接收者：父界面类对象

信号为子界面emit的自定义信号(可以只声明不写定义)，其中在子界面内部"main函数中"connect

好按钮与自定义的信号：

connect(点击子界面的返回按钮clicked , emit自定义好的信号)

槽为lambda函数，hide子界面再show父界面

神奇的是，只要在父子界面类对象定义完成后，这一去一回两个connect的先后顺序可以颠倒，没有了执行逻辑上的先后关系，很像是积木中把两个元件连接在一起的感觉。

用合适的数据结构封装关卡内的怪物进攻路线，我使用的是QVector<QVector>,location是我自己封装的坐标结构体，内部一个QVector就是一条路，外部QVector装载这个关卡的所有路径。

为方便通过关卡数来获得这一关的路径集，我在global_use.h中封装了一个Paths类，在里面把每一关的路径写死了。

怪物由于有共性（血量这个变量、走相同的路径集、走到终点时向GameWindow发信号使生命减少），又有不同点（不同的绘制图片路径、血量具体取值不同），故采取继承的方式来创建不同类型的怪物：怪物基类是共性的地方、派生类是不同种类的怪物。

防御塔同理，不过不同的是，近战塔和远程塔的放置位置、攻击效果有较大区别，故对近战塔和远程塔分别构建一个基类，针对不同的近战塔派生出具体的近战塔类。

基于QMouseEvent类实现对鼠标事件的捕捉。重载mousePressEvent函数捕捉对鼠标点击像素区域，没有使用按钮

创建防御塔的初始思路是将地图上特定的砖块位置放置按钮，按钮可以触发选择防御塔的按钮，但是按钮生成按钮与隐藏按钮较难实现

故最终选择了捕捉鼠标位置信息来显示选择防御塔按钮的方案。

游戏运行主体的实现思路：

通过设定一个定时器，定时器每个周期相当于游戏的【一帧】，每一帧调用敌人类中的移动、防御塔类中的攻击等，完成游戏循序渐进的运作。

游戏进行的每一帧中，伪代码如下：

```
    游戏内防御塔和敌人的运作
    遍历每个防御塔
        遍历每个敌人，将防御塔攻击范围内的敌人收集起来 送给该防御塔 防御塔对这部分敌人发出攻击
    将birth阶段的敌人移动到on_stage阶段中
    遍历每个敌人
        敌人移动
        判断敌人是否【移动到终点】或【血量减到0】，若满足其中之一就将其从on_stage阶段移除
    判断胜利
    重新绘图
```

为了方便判断游戏是否结束，同时方便绘图，将敌人分为birth阶段/on_stage阶段（用两个vector装两种阶段的敌人对象），生成敌人时只在birth数组内添加敌人，在游戏开始后每一帧内，都将birth阶段的全部敌人输送给on_stage数组，在敌人走到终点或敌人死亡时就将其从on_stage数组中移除。

如此一来，游戏结束的标志就是birth和on_stage均为空，且玩家生命>0时游戏结束。

绘图函数把on_stage的敌人全部打印就行，只绘制“在场上”的敌人即on_stage中的敌人对象。

3.实验过程中遇到的BUG:

BUG1：只修改头文件再编译运行，程序好像还是修改前的样子

程序依赖于.o,而.o依赖于.cpp,故假设gamewindow.cpp include了 global_use.h 虽然更改.h后.h会重新编译，但是编译器并不会重新编译.cpp，故看起来属性都没变，实际上是.cpp没有重新编译的缘故。

目前的解决方案是随便在gamewindow.cpp中加入一个int，在更改完.h后修改下int值，造成.cpp文件的更改，从而使.cpp也重新编译。

BUG2: (将来的隐患bug)

gamewindow.h中, 可以预见加远程塔的过程较繁琐:

【选择远程防御塔界面】及【远程塔的升级移除界面】在鼠标事件处理里每一个其他的界面点击时都要hide, 故整个鼠标点击事件要检查一遍, 把需要远程塔界面关闭的地方全部加上。

BUG3:

在防御塔的Attack里写防御塔被怪物攻击的功能, 无法把OnStageVector中的对象实体中的血量减少。

分析BUG3的原因:

可能的几个问题:

1) towerbase.h中没声明虚函数, 可能在tower_1.cpp中没有override成功

2) gamewindow.cpp中第357行的循环, 如何将引用值从OnStageVector传到防御塔的Attack中从而对OnStageVector中的ENEMY对象造成真实伤害?

3) 时间切片的问题, 我目前设置成每隔一秒地刺才会对范围内的敌人发起扣血功能, 但是由于一帧100ms敌人会移动10像素, 所以可能不到1s敌人就会移出地刺所在的block(60像素), 所以敌人过快了。

新的构想 (实现真伤): Attack把OnStageVector中的敌人对象拷贝过来, 把血削了返回, 再在OnStageVector中用这个返回对象的属性覆盖到原对象中。

BUG3问题解决: 原来是因为没有在TowerBase基类声明Attack为虚函数, 然后再在Tower1中override, 导致Attack执行的一直是基类的attack, 就什么也没做, 不是值传递的问题。

4.有趣的地方

设计了按钮的点击特效。我根据在B站学Qt(URL:<https://www.bilibili.com/video/BV1g4411H78N>)中讲到的方法, 设计了主界面和关卡选择界面的自定义按钮类MyButton, 从而可以实现摁下按钮按钮弹一下、以及通过切换图片的方式来实现按钮下陷和凸起的不同状态, 提升了点击按钮时的视觉体验。UI的体验优化非常有趣(且不太需要debug哈哈)。