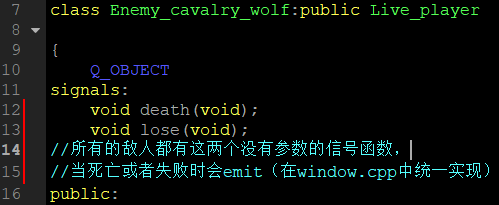
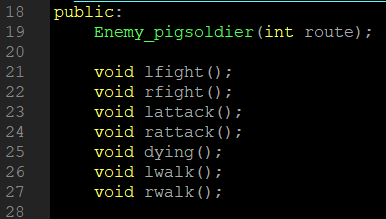
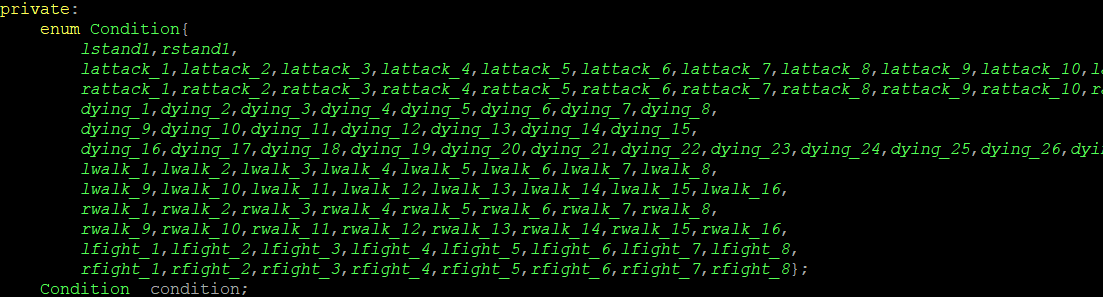
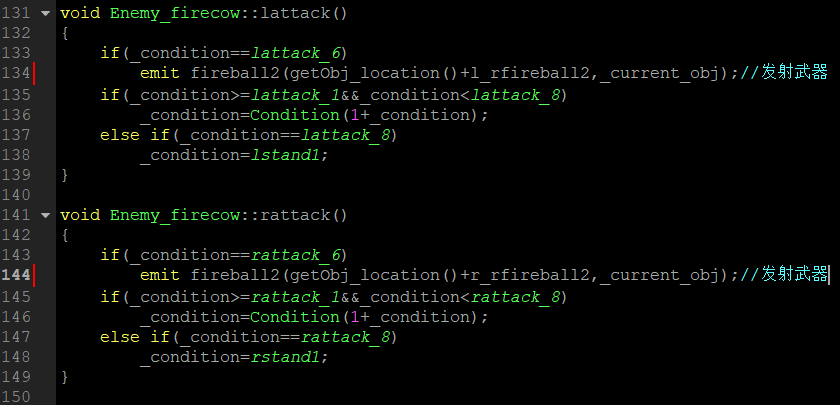
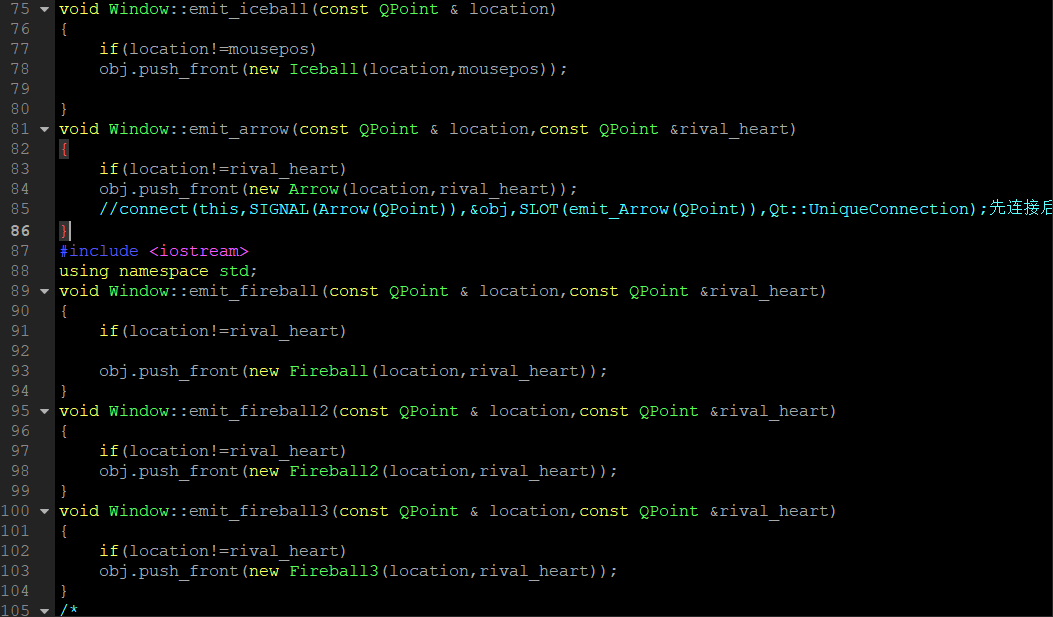
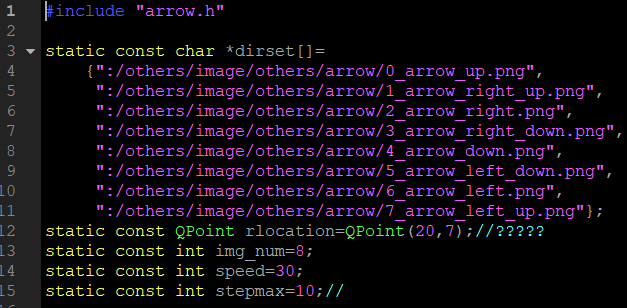
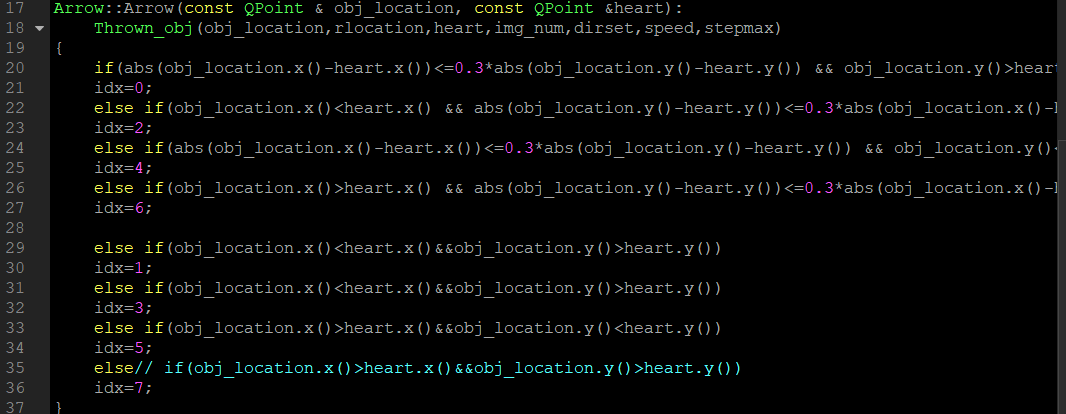
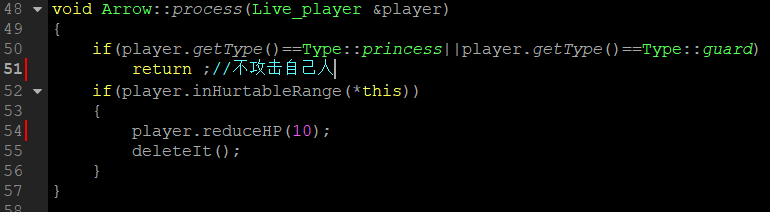
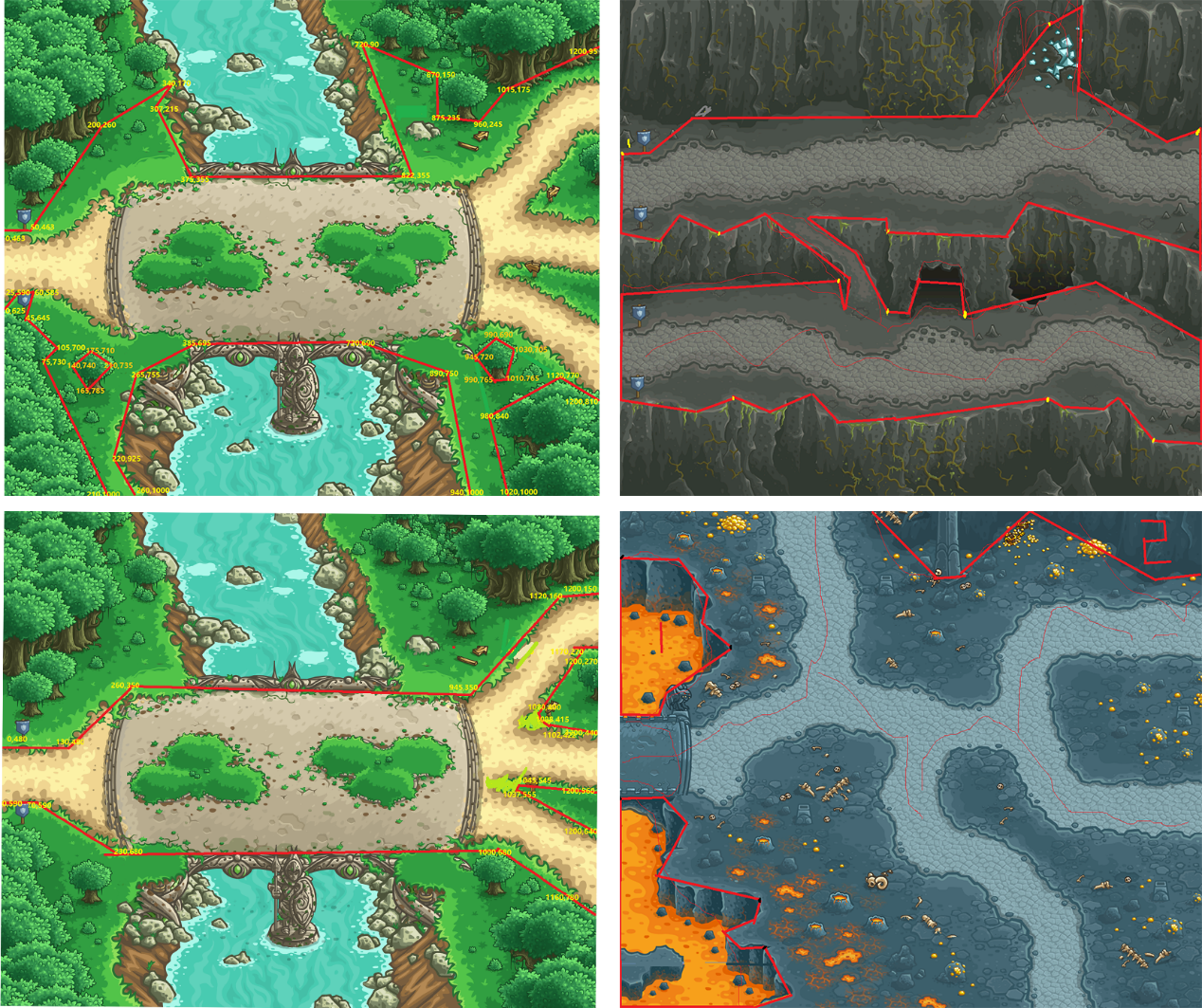
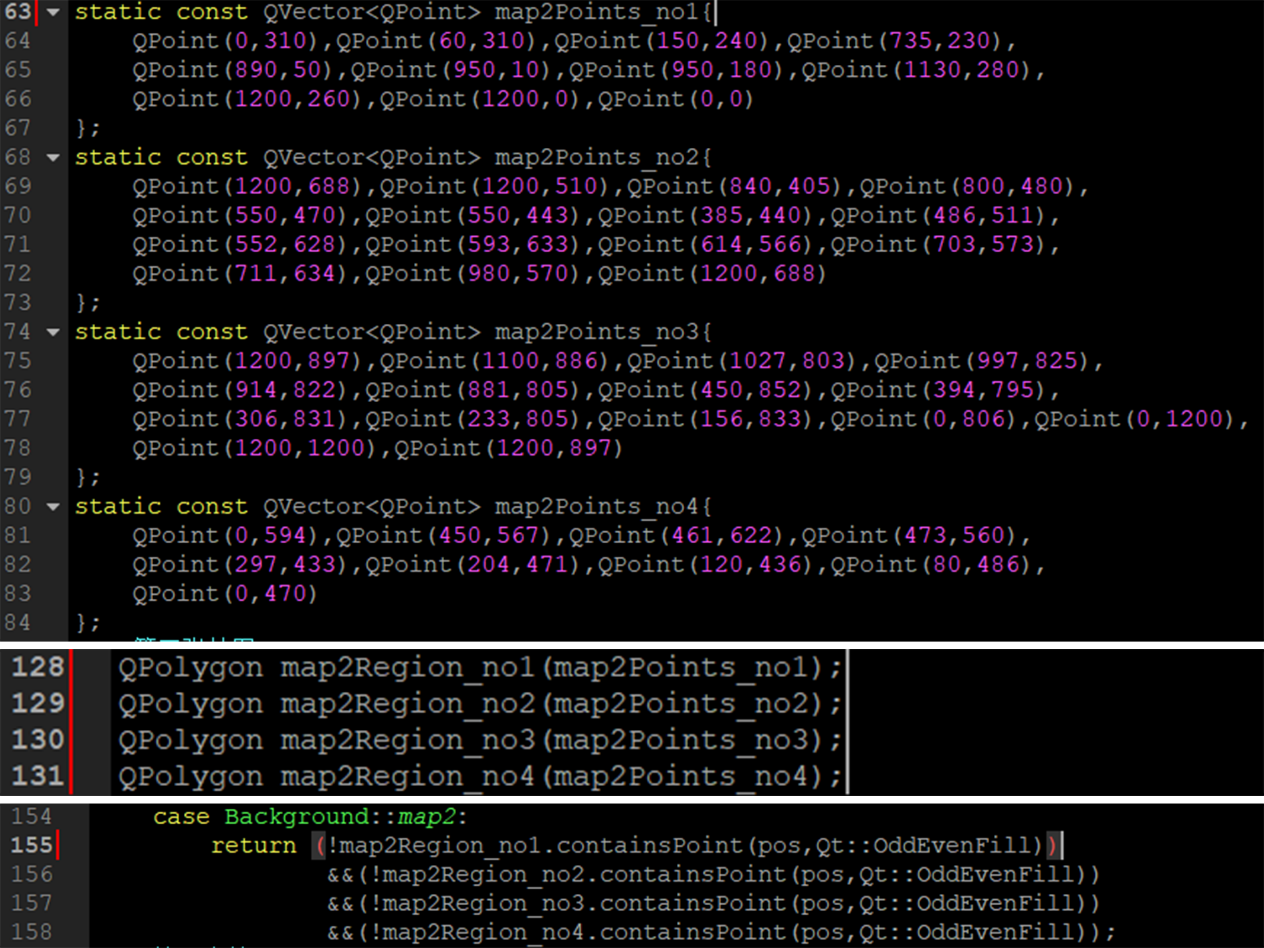
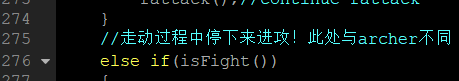
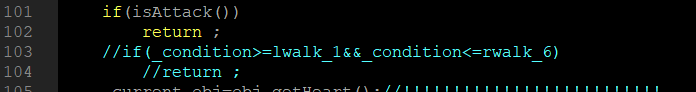
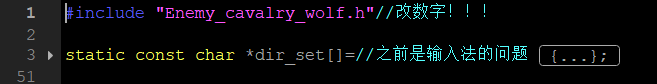
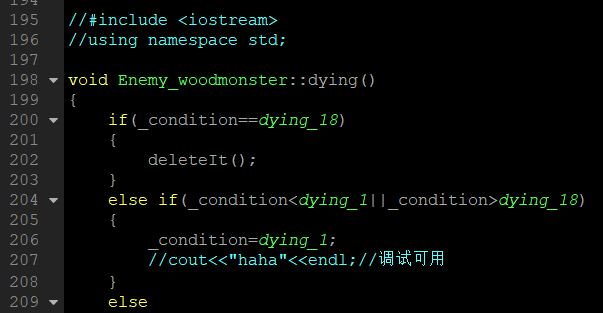
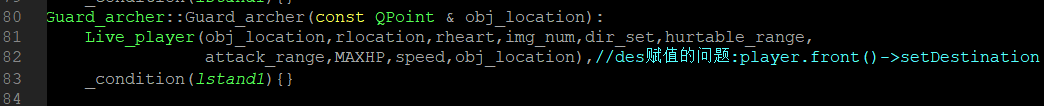
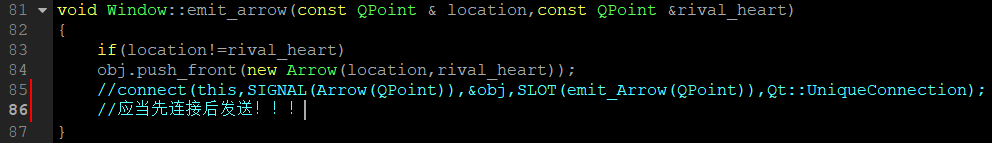
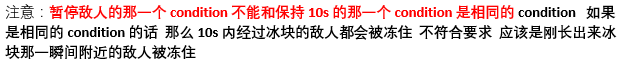
1. （具体描述自己分工部分的实现方案）我的工作是一下几个方面（合计**16.5**个类，外加图片处理）;
   1. 人物及对应的武器，共13个类：
      1. Guard\_archer与Arrow
      2. Enemy\_wolf
      3. Enemy\_firewolf
      4. Enemy\_cavalry\_wolf
      5. Enemy\_firewizard与Fireball
      6. Enemy\_firecow与Fireball2
      7. Enemy\_pigsoldier与Fireball3
      8. Enemy\_woodmonster
      9. Enemy\_spider
   2. constant.cpp中的map1,map2,map3的地图规划，即constant类的部分实现。
   3. 冰女的三个技能，共3个类：
      1. Icecone
      2. Icefrozen
      3. Lightning
   4. 处理图片，包括：
      1. 对话框
      2. 黑白渐变
      3. Button、冲击波等等的颜色替换
      4. 各种批量改图像或者画布的大小
      5. 各种批量左右翻转
      6. 合并图片
2. 核心代码及注释：
   1. 人物部分：
      1. 
      2. 是我的风格，与另外两位成员的react函数左右相同，均是改变\_condition的（\_condition本质上代表图片的编号）。
      3. 这是\_condition的种类，一个\_condition代表一张图片，有些图片共同代表一种行为，如死亡（**dying**），向左、右远程攻击（**lattack\_/rattack**），向左近身攻击（**lfight/rfight**）,向左、右行走（**lwalk/rwalk**），向左、右站立（**lstand/rstand**）等等（在基类中写入了这种数据成员与图片的关系如何对应）。
      4. 判断\_condition的状态.
      5. 均是用来减缓运动.
      6. 
      7.   
           
           
         有武器的人物及emit武器的函数的实现。
   2. 武器（仅以Arrow为例）：
      1. 载入图片。
      2. 跟据敌人的相对位置判断箭头的方向，共有八个方向。
      3. 攻击敌人（即enemy）。
   3. constant.cpp中的走动范围：
      1. 改了好几次的地图（上面标的有坐标，用黄色字体显示）。
      2. 这是constant中map2部分的代码。
      3. 原理：以map2为例，利用Qt库中的类QPolygon构造几个多边形，再用QPolygon的成员函数containsPoint来判断点是否在该矩形内部。
3. 遇到的问题以及解决：
   1. 写的一个类是Enemy\_wolf，总是不能运行，debug了两天才发现是没有按照图片个数改img\_num，导致越界。
   2. 因为擅自为了“让减血效果更明显”而更改了hurtable\_range和attack\_range的值，导致武器刚一发出就deleteIt()而不能发出，又de了半天bug才发现。
   3. l\_rfireball2和r\_rfireball2指武器与人物的相对位置，它们的值保证武器能在正确的位置发出。
   4. 将process函数的参数的const属性去掉后，使reduceHP更加方便，而不需要再添加很多东西（是我在写Enemy\_pigsoldier（因为开始设置了近身攻击，而不是像之前那样用武器来减敌人的血）时建议去掉const属性的，为了代码的简洁性）。
   5. 一些bug的修复：
      1. 
      2. 之前条件判断错了。
      3. 
      4. 将作用相同而冗杂的代码化解（代码冗杂现象在不停的debug中十分容易出现，因为在debug中要尝试很多地方的改动，最后甚至会改变原先的代码思路）。
      5. 因为输入法问题，导致图片地址错误，结果屏幕上不能显示图片。（我的电脑上有三个输入法：法语智能输入法、搜狗拼音输入法和微软拼音输入法。法语智能输入法的下划线字符“＿”和另外两个输入法的下划线字符“\_”不一样。。。）
      6. 在process中只进行不同阶段之间的改变（例如lwalk变为lattack\_1），而不进行同一阶段的连续（例如lattack\_1变为lattack\_2），解决了只能判断一个敌人的问题。
      7. 万能的输出调试。
      8. 在构造函数只有一个参数（即初始位置（obj\_location）的坐标）的情况下调整目的地（destination）坐标的方法：模板数组的库函数.front()来取出最新加入的成员。
      9. 把辛辛苦苦写的Shockwave类删掉了。。。（因为冗余）。
      10. 删掉了能够上下走的狼（因为画面太鬼畜）。
      11. 刚开始因为没太懂槽与信号函数的连接机制，所以connect的使用时间判断错误。。。
      12. 组长在任务分配里就提示得很好了，预防了一个大bug点个赞！！！！！！
      13. 写冰女的三个技能时，我用了两个static成员变量，（并用idx%=26来更新并防止其值过大），后来因为这样不太好所以删掉了，改成了普通的私有变量。
      14. 刚开始写代码是能只public，就不private（嫌麻烦），也几乎不用const、引用&什么的，后来开始慢慢在意了。
      15. 出现.qrc资源图片的问题就qmake
      16. Qt编译器自身的问题;
          1. Could not find qmake configuration file default. Error while parsing file（打不开工程）
          2. Error while building/deploying project (kit: Desktop Qt 5.8.0 MSVC2015 32bit) When executing step "qmake"
          3. wordpress out of memory（设定的内存不够，需要加一句话来改变设定）
          4. error: cannot declare field 'Window:: ' to be of abstract type
          5. [release/qrc\_resource.cpp] Error 1:File not found
          6. 等等，等等

万能解决方法：百度及谷歌，中文或英文。

1. 总结与收获：
   1. 前不久看知乎，看到一句话说，程序员有两种：一种是真正在搞东西的大佬，是少数；另一种是在别人的框架下写东西的却是多数。想成为组长那种少数人，就得有极其强大的短期自学能力和极其强大的耐心，才行（而不能总靠别人给自己讲新知识QAQ）。
   2. 编程需要足够足够足够足够足够多的耐心，因为单独一个大作业就可以把2学分上成10学分（皮这一下我很开心^\_^）。
   3. 完成一件任务的过程中可能会做很多无用功，有时是因为没理解好任务的意思，有时是没想清楚代码的思路，有时候，这些无用功却是必须的，是为了让其他的功变成有用功（为什么我开始煽情了QAQ）。
   4. 刚开始以为自己绝对写不出来，单是看懂组长写的框架就花了整整两天，然后又花了整整两天写第一个类Enemy\_wolf。之后再写就好多了，但是虽然各个类的整体思路是一样的，但是因为功能的一些小差别而使bug次次都不同。。。
   5. 学会了Qt的信号与槽机制，真正会用paintevent、pressevent等了。