高级程序语言设计项目报告

报告人：朱霞洋 学号：2113301

1. **作业选题**

模拟扑克游戏

1. **开发软件**

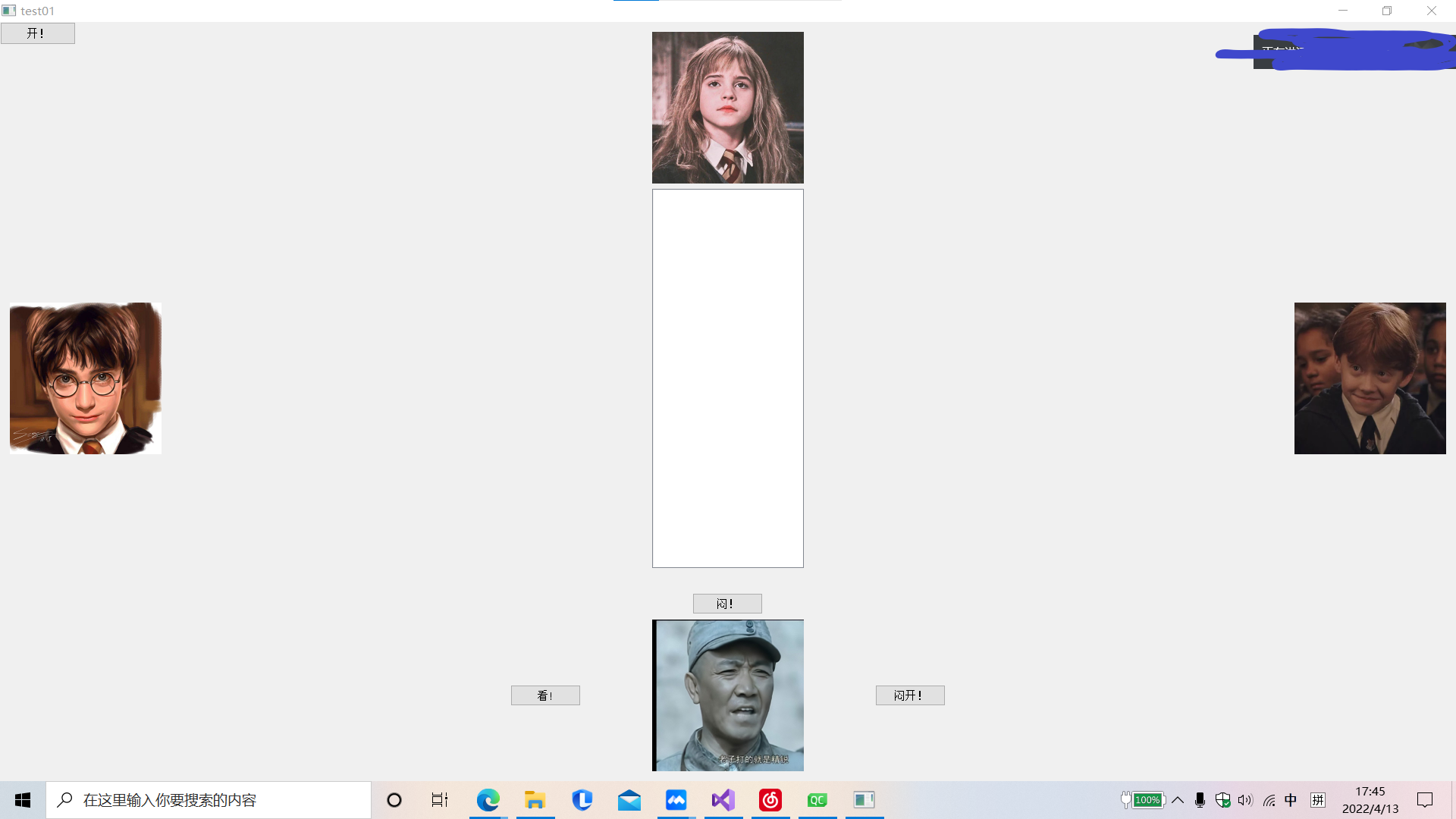
Qt 5.14.2

Qt Creator 4.11.1

1. **作业要求**

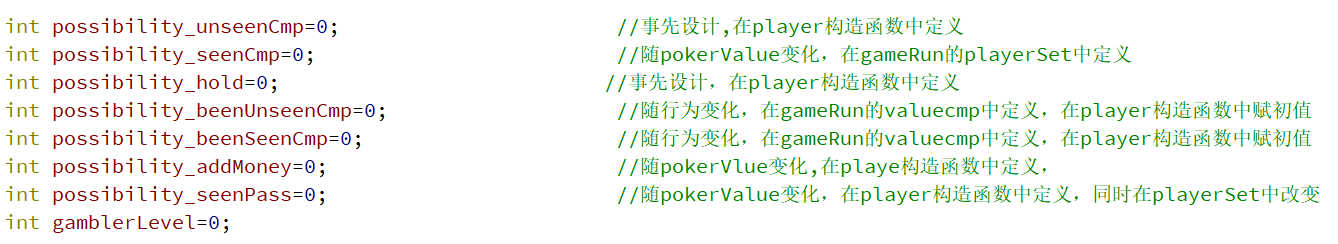
运用所学知识（面向对象等）完成图形化编程，内容主题不限

1. **整体开发流程**
2. 通过自定义类完成游戏开始界面，关卡选择界面，以及游戏进行界面的大致设计；



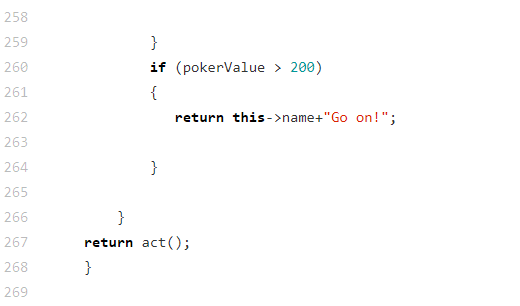
1. 构造牌类Card，完成牌的初始化，洗牌等操作；

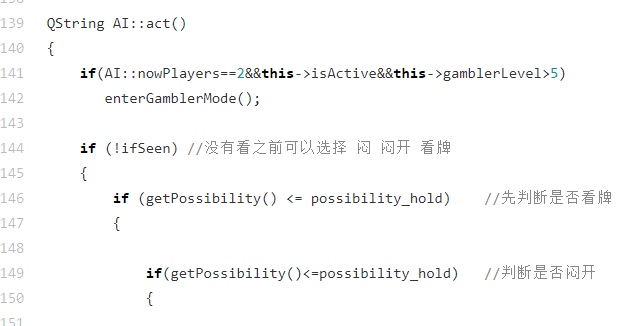
（1）牌类是整个开发过程中变化最少的一个类，基本在写好后就没有改动，稳定性良好；

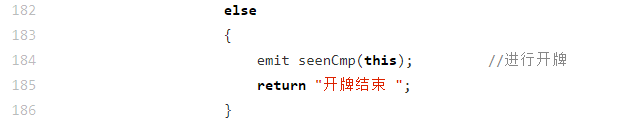
1. 构造一个AIPlayer的基类，为ai玩家赋予看牌，弃牌，加注，比牌等操作行为；
2. ai的行为逻辑是整个游戏开发最不好设计的部分，尤其我想做到为每一个ai制定不同的行为风格，于是在AIPlayer基类中制定了很多属性

思路如下：

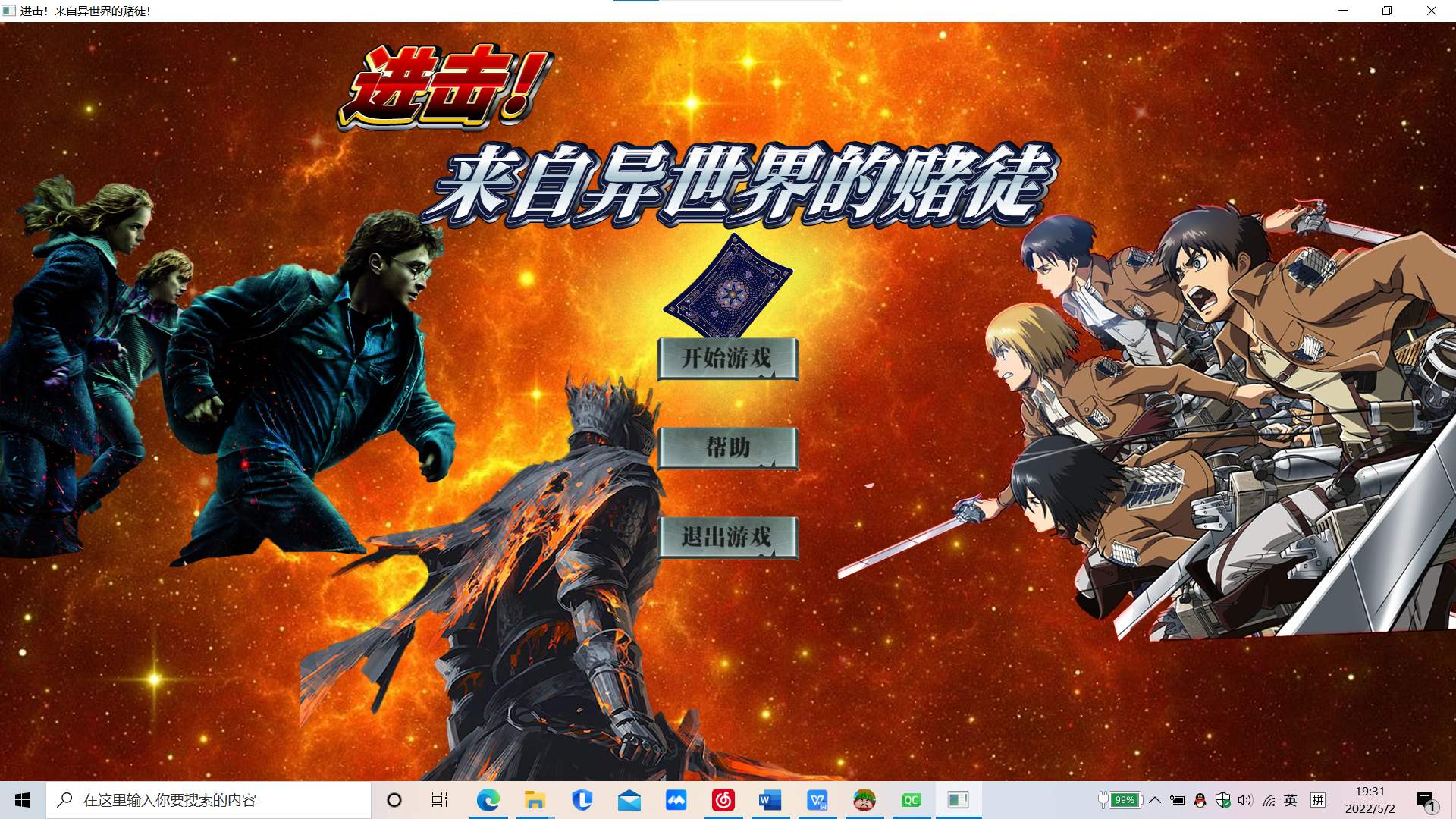
每个ai创建时会自定义他选择加注，看牌，弃牌，比牌的概率和一个gamblerLevel，如比较胆小，则看牌概率等较大，gamblerLevel较小，胆大反之；并且随着游戏进行，这些属性会根据场上人员的去留，比牌结果进行相应变化，且变化标准并不相同，与该ai的gamblerLevel有关，从而实现每个ai有一定的游戏智能和不同的游戏风格。

1. 其中最重要的是一个act函数，ai玩家根据场上情况和手牌大小，个人属性，选择要进行的操作，看牌，弃牌，抑或加注

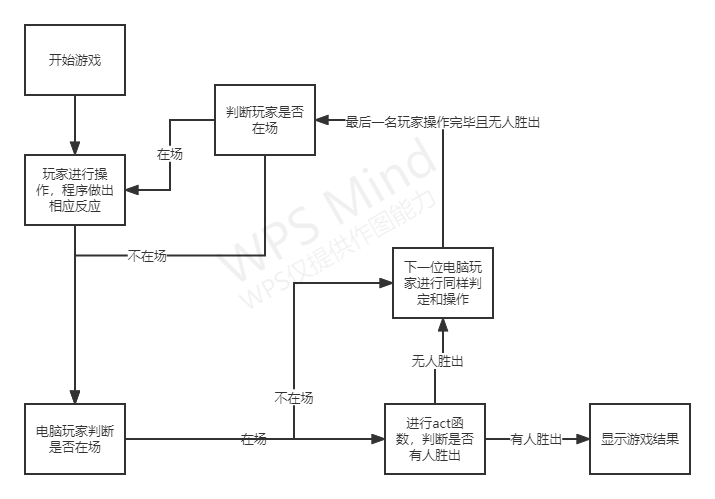
在最初版本中，总共写了150+行来实现，而在最终版不仅代码量大大减少，效果也比最初好上许多，不再是单一标准的评判



1. 细化游戏逻辑，连接相应的信号和槽函数，游戏至此能够运行；
2. 增强游戏性，用类的继承来写每一个ai玩家，使其有不同的行为风格，同时赋予每个ai玩家独特的语言包，增强游戏趣味；
3. 一些细节优化，如自定义myButton类来实现按钮的不规则形状，鼠标按下和悬停的特效等；
4. 开发最后阶段优化游戏的内存占用，设计每个场景的ui，bgm；

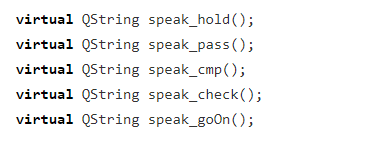


1. **游戏运行框架**

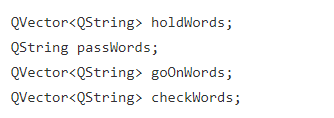


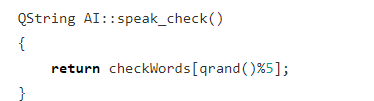
1. **项目开发重点与难点以及亮点**

**难点：**

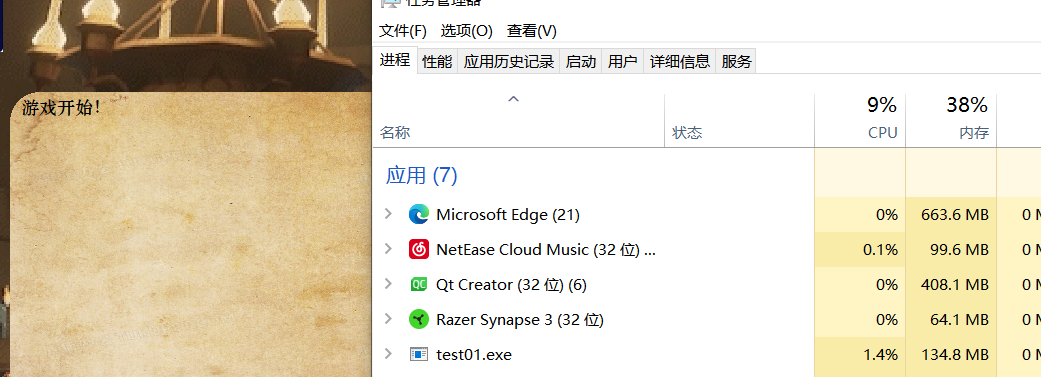
1. 对AI行为的编写，如上所述，主要运用类的派生，和各种判定以及属性赋值来解决；
2. AI的个性编写，该游戏会涉及每个AI角色不同的语言风格，最开始是在基类中定义虚函数，之后在派生类中重写该函数解决

后来发现这并非最好方案，因为如此做法会需要每个角色的派生类都要重写一遍函数，工作较为重复繁杂，于是进行改进

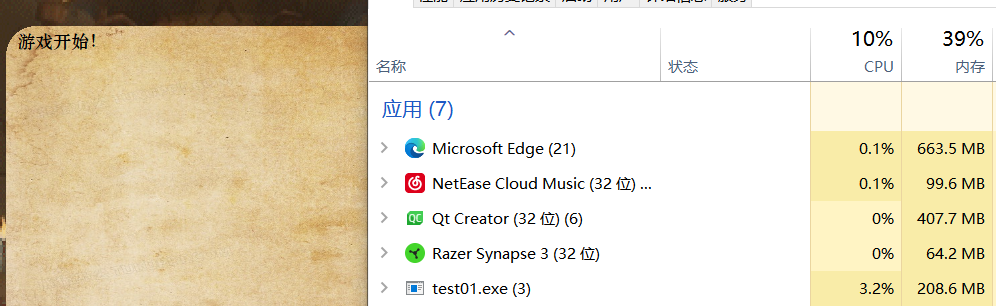


用这样的vector容器可以方便的在每个派生类的构造函数中就定义好该角色的语言包，然后调用基类函数起到同样的作用，减少了工作量的同时更加直观方便。

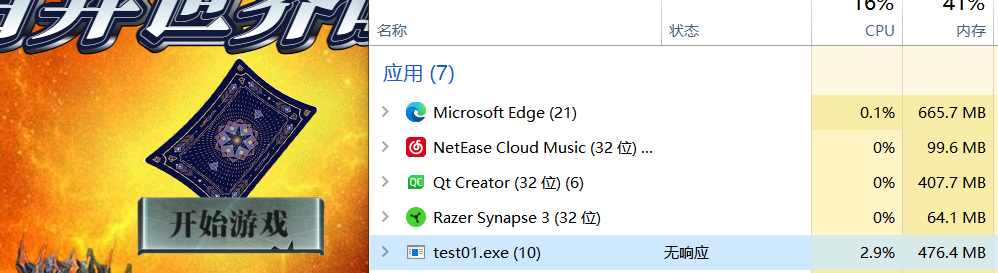
**重点：**

1.对内存管理。最初功能比较简陋时并没有想到对游戏内存进行管理，直到开发末期测试时发现在反复加载游戏场景后程序会越来越卡顿，发现是内存占用过大导致

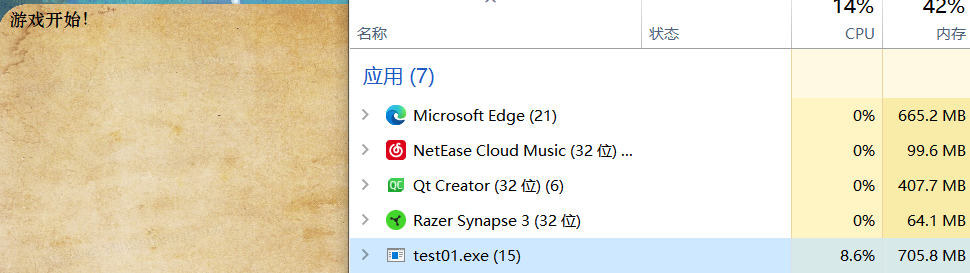
可以看见最初时游戏只有134mb的内存



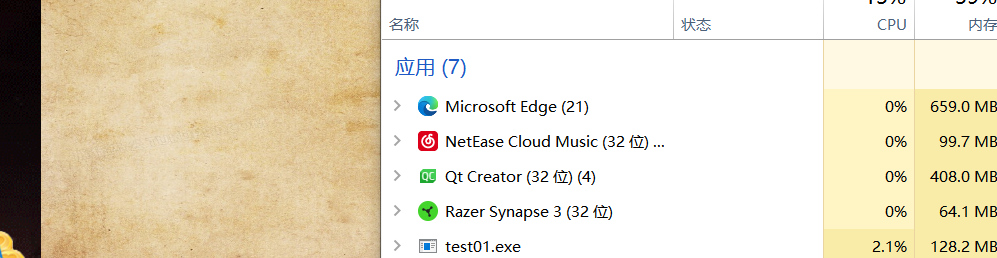
刷新界面若干次后来到了208mb，此时仍能流畅运行

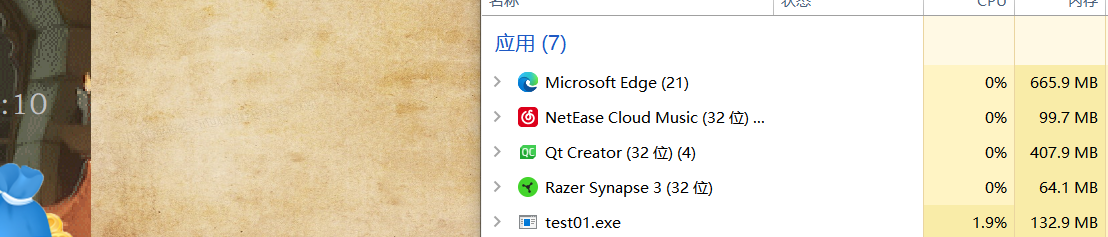


即便是返回了界面主菜单内存仍没有释放完毕



为了测试程序效果刷新了十多次，内存直接到达705mb，此时卡顿明显，游戏体验明显下降；在对内存管理进行优化之后，内存占用稳定在130mb左右，即便刷新界面十余次仍没有太大变化，这是项目开发中的一个重点改进内容。此前只是粗略学习动态内存分配，对于其作用和意义理解不大，经过这个项目的改进，终于有了深切的体会。



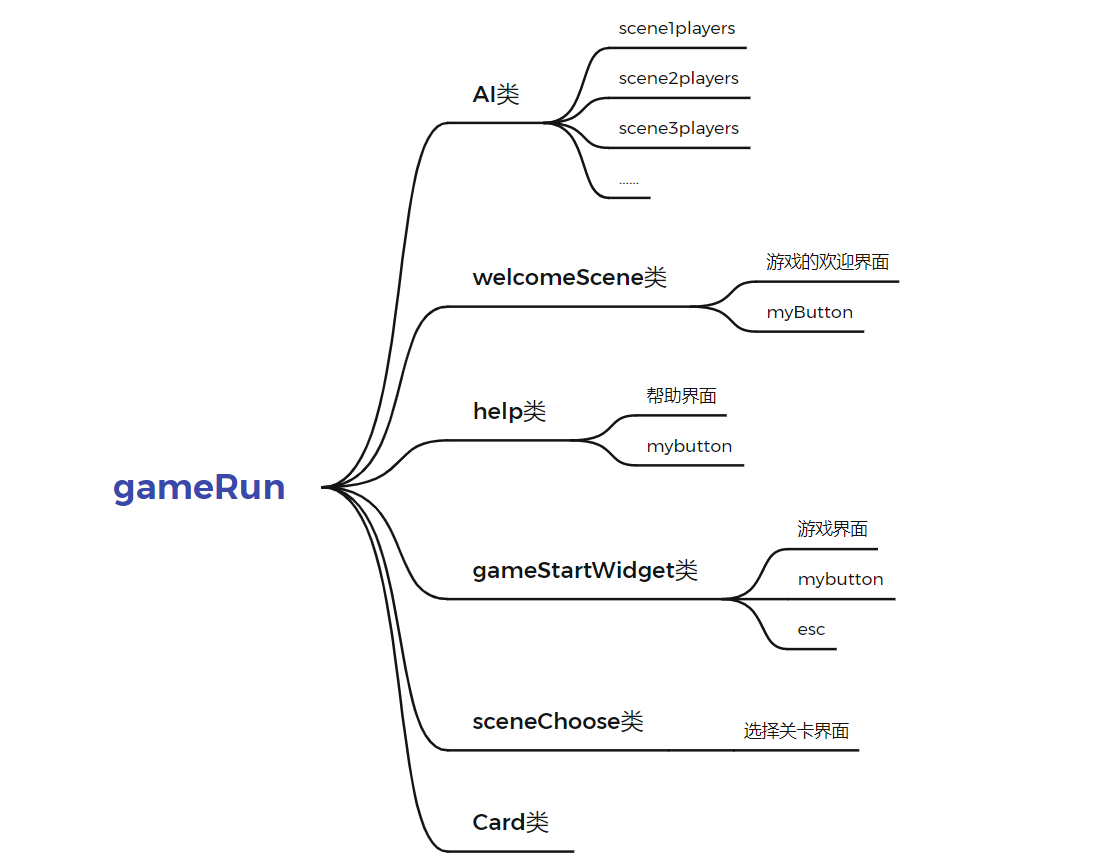


1. 资源文件的管理。本次项目用到了大量的资源文件，发现当某个资源文件过大，或是资源文件过多时程序会无法运行，且构建非常缓慢，最终在工程文件中加入CONFIG += resources\_big该指令得以解决。
2. 实现按钮的鼠标悬停，按下特效，qlabel的点击。运用事件过滤器和自定义myButton类实现该效果。

亮点：

* + - 1. AI部分的编写；
      2. 特效的设置
      3. 多元素的融合；

1. **具体实现**
2. **主要的自定义类以及其关系**



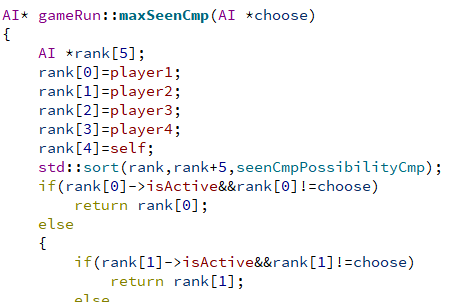
1. **一些算法**
2. 大小判定



1. 根据牌的大小二次更改属性



（3）寻找应该被比牌的玩家



1. player的act函数，同上
2. **信号与槽的连接**



1. **可以改进的点**
   * + 1. ai仍然有优化空间；
       2. 流程很短，如果从一个游戏的角度来看故事性不足；
       3. 暂时没有存档机制；
2. **收获**
3. 对于动态内存分配以及指针等有了更深的认识；
4. 虚函数，析构函数，STL的运用；
5. 长代码的编写能力，图形化编程能力有提升；
6. 个人通过查阅资料解决问题，实现效果的能力；
7. Ps能力的提升
8. 成就感