

第三次课程设计报告

课程设计的主要内容、目标及设计思路

- 主要内容和目标：

主要设计准备做一个类似于水滴清单的一个任务清单类型的软件：主要功能就是能够对于不同登陆的用户选择适当的已经存储好的清单，并且能够对于用户选择的任务进行倒计时，同时还可以选择性的开启学霸模式（屏保开启，保证不分心），否则还可以选择开启音乐播放器，能在做任务的同时听听音乐，音乐可以选择导入默认的音乐清单（具体有愛にできることはまだあるかい等等），还可以导入自己的默认音乐清单。完成任务后有系统提示和音乐提醒，还有轻松时刻，可以打小游戏放松一下/捂脸。

- 设计思路：

准备事先将每个部分各自实现好，然后因为都是从QWidget下继承下来的窗口对象，可以相互之间构造父子窗口关系/发抖。提前将每个窗口写好后，可以通过信号与槽的机制，emit出一个传入对应窗口的信号，将每个有关系的窗口相互联系起来。

类的设计以及类层次的关系，用到的OOP设计思想

设计中的Loginwindow和scheduleWindow都是继承自QMainWindow的，况且它们之间有相同的函数，例如绘制背景的函数等等，故由此设计继承关系；然后LoginWindow里面要不断的根据不同的用户而不断的创建和销毁对象，还设计了一个user的类，用来实现用户的登陆，故使用指针指向scheduleWindow，并同时emit ToScheduleWindow这个信号，将即将登陆的这个用户的名字传送给即将生成的这个Window，形成了信号与槽的连接。

对于ScheduleWindow而言：

设计了一个Notes类，专门用来存储每一次的note，以及相对应的内容的判定。

还设计了一个MyMenu的类，继承于QMenu，便于ScheduleWindow中的使用。

里面有不同的stackWidget在其中，这个stackWidget是qt已经封装好的了，我拿来直接用作用户不同操作的转场。这些stackWidget均以ScheduleWindow作为parent，实现预期完成的功能。

对于音乐盒子窗体而言：

我设计是音乐能够独立于schedulewindow之外自己播放的。故仅仅是单独设计了一个窗口。然后在里面另外设计了窗体的格式，使其变得更加人性化，重写了函数如下：

```
bool nativeEvent(const QByteArray &eventType, void *message, long *result);
void mousePressEvent(QMouseEvent *event);
void mouseMoveEvent(QMouseEvent *event);
void contextMenuEvent(QContextMenuEvent *e);
void mouseReleaseEvent(QMouseEvent *event);
```

实现了该窗体的移动和一系列功能。

对于打砖块窗体而言：

类有：ball,brick,knock分别为球，砖块与击打板，每个对象都是用QRectF进行存储当下的所占的矩形，然后都有重新设置自身状态的函数以及获取自身状态的函数。

类层次关系不太明显，主要是中心的mainWindow核心游戏控制类不断地与ball,brick,knock发生不断的数据交换

程序的功能特点和运行操作方法

功能特点：

是各自相对应具有各自功能的部分都是可以直接抽象出来的，各个对象之间的耦合度不是很大，具有较强的拓展能力，还可以从这个窗口直接继承下来进行二次开发。

运行操作方法：

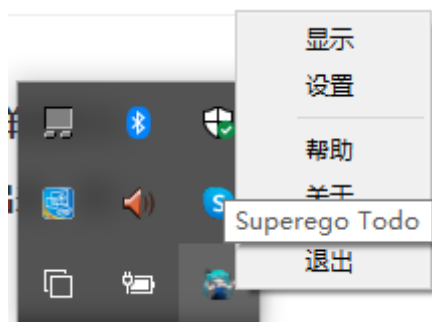
具体是这样的：

设计登陆的窗口，然后是每个人各自的用户密码登陆(有用户名:abc密码:123)用户的密码是通过M5加密算法得出来写入文件的，确保了密码的安全性。

(密码错误5次以上就会自动封锁账号)



成功进入后，会自动生成托盘程序（备注：按键只有显示和退出有效/捂脸），这是todo清单的程序窗体，可以实现增加代办，完成代办等一系列功能，还能记录完成的次数

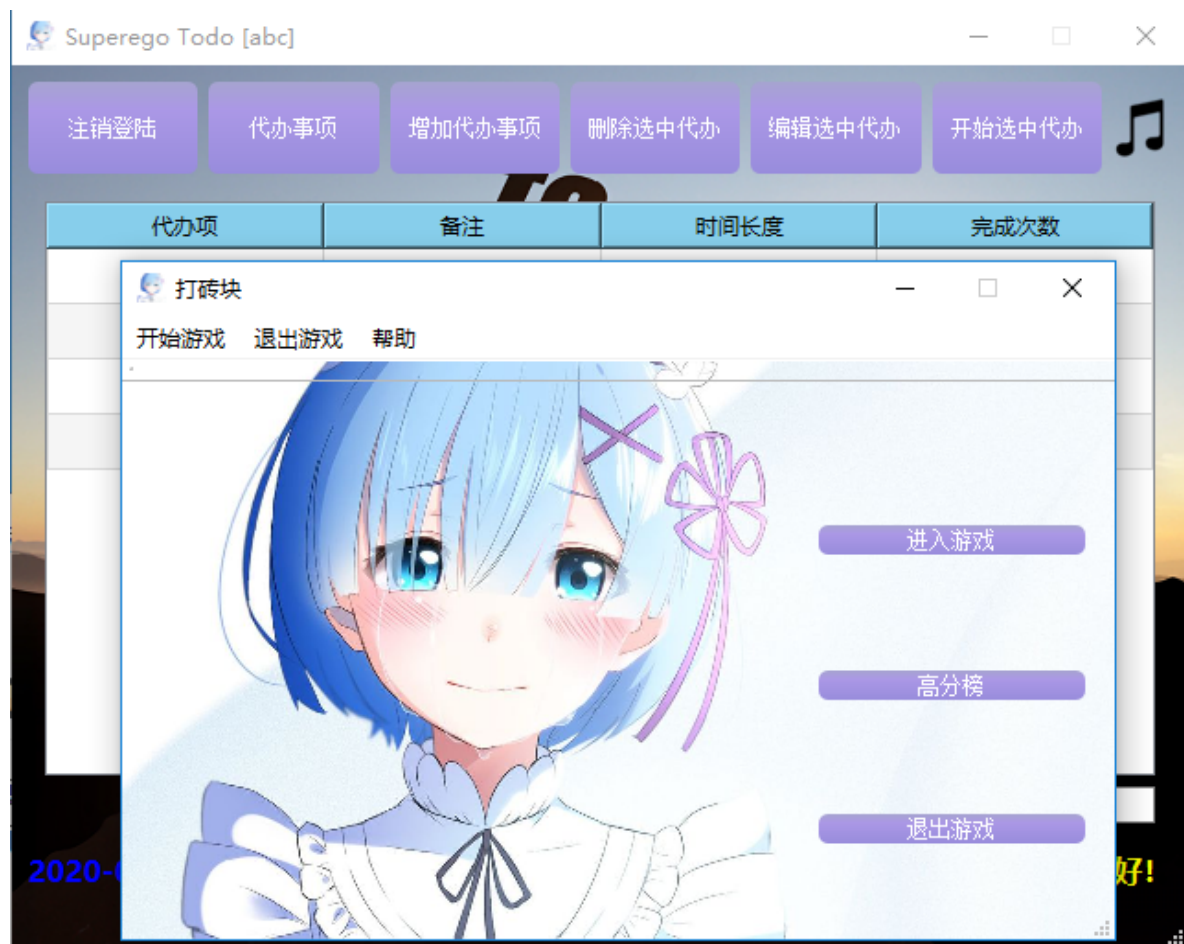




然后就是音乐播放盒子了，可以实现音乐清单的自动导入和清单上面右击删除音乐，双击播放音乐，frame上右击可以实现下一首和更换背景图片。（具体更多的功能请将鼠标移动到对应的图标上会有 tooltips 的噢。



最后是打砖块游戏的实现，具体的细节可以查看打砖块游戏窗口上面有帮助文档。



遇到的问题 and 解决方案、引发的相关思考

遇到的问题：

与之前的两次用控制台写出来的程序不太一样，这次用qt写遇到了许多与window下不太一样的地方：

①qt设置QPushButton的时候，图片是自动显示成为不缩放的，然后图片在按钮上显示不出来，我一度以为是qt无法加载png类型的图片，最后将按钮放大才发现的

解决方案：

```
ui->helper1->setStyleSheet("border-image:url(://res/helper1.JPG);");
```

②qt在建立类的时候会帮你自动继承某一种比如QWidget，但是我实际需要选择的是 QMainWindow，因为QWidget下无法生成菜单栏，所以当我new一个菜单栏的时候会发生报错提醒。

解决方案：在初始选择的时候，选择从QMainWindow继承

③在设置屏保的过程中，遇到程序总是异常结束的问题，经网络查询得知，在使用qt中的window的某些api时，会引发win自身系统的查杀病毒的defender的扫描，造成文件损坏。

解决方案：利用vs先将屏保程序抽象成为exe文件，再使用qt中的QApplication下的成员函数本地打开exe文件即可解决问题

由课设三引发的思考：

这次我做的工作量相比前两次课设而言都大大的减少了（当然也有可能是打砖块这个游戏比较简单的缘故），所以如果使用已经提前封装好的软件比如qt，或者是已经设置好的应用框架的话，做出同一件成果的工作量要大大减少。