# 第五周实验报告

这周主要完成了以下工作：

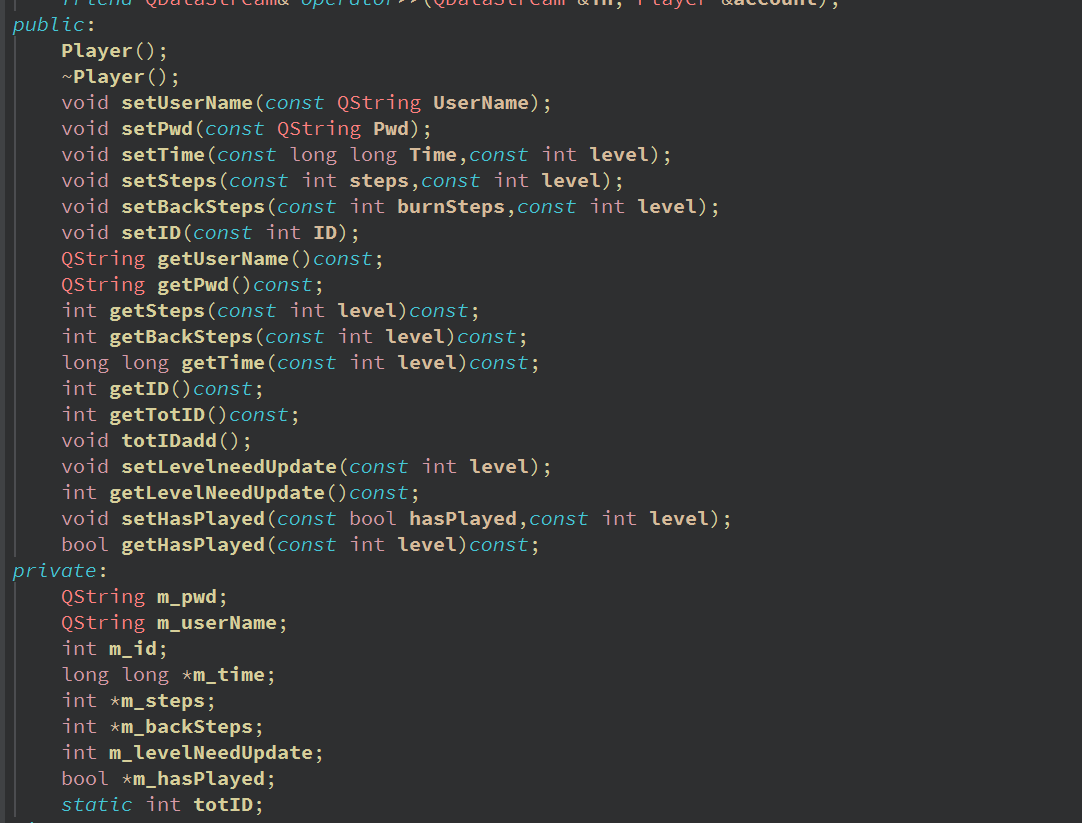
1、设计了一个用户类，用来记录当前用户的所有相关信息，包括密码、用户名、游戏记录等。

2、简单设计了登录页面和注册页面，能够让玩家登录和注册账号。

3、实现了对用户信息的文件管理，并且实现了对当前玩家的游戏记录的实时更新。

下面详细介绍一下各个工作的内容。

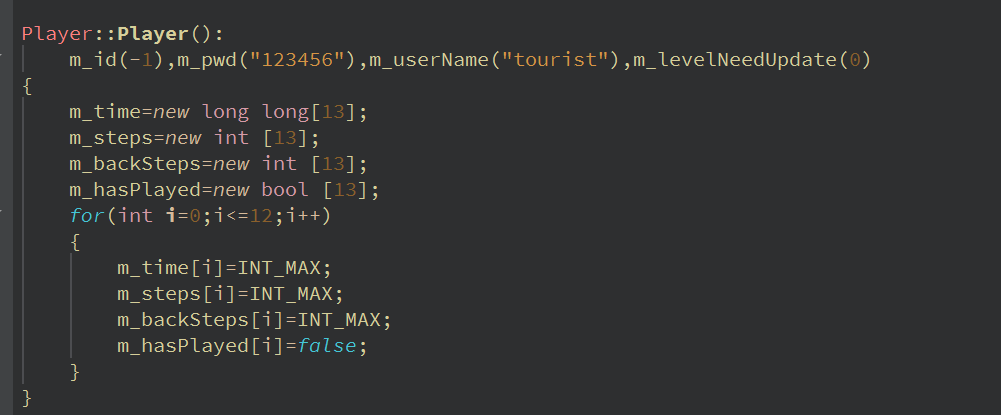
## 用户类的设计



用户类包括了以下内容：用户ID、密码、用户名和每个关卡的相关游戏记录。

每个关卡的相关游戏记录又可分为：是否游玩过该关卡(m\_hasPlayed)，通过时间(m\_time)，花费的步数(m\_steps)，花费的撤回步数(m\_backSteps)。另外对这些数据成员添加了一些接口，实现了对这些数据成员的修改和访问。

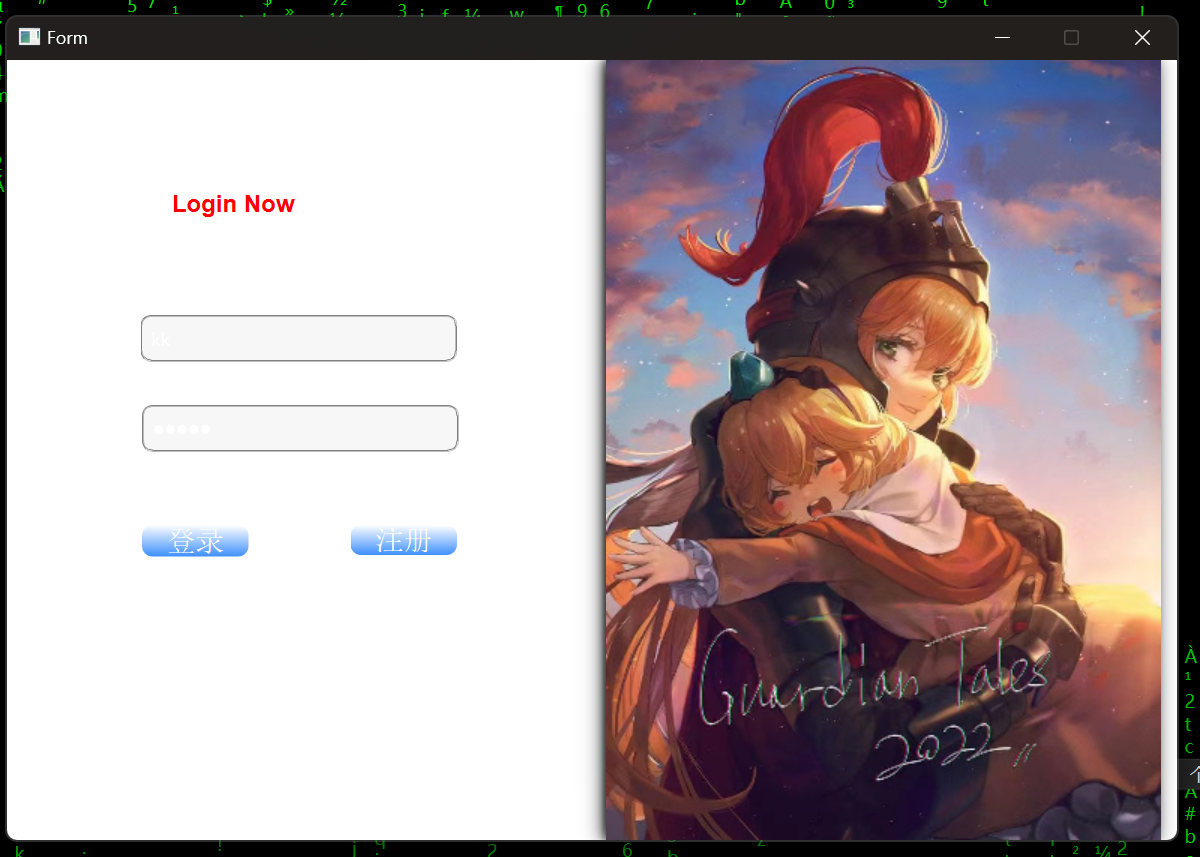
同时还添加了一个m\_levelNeedUpdate，后面对实时更新玩家的游戏记录有很大用处。



这里的默认构造函数将Player的当前用户信息设置如上，作为游客模式，游戏本身不会对游客玩家的游戏记录进行保存和更新。

## 登录页面的设计

设计了一个登录的页面：



下面就是对应的相关逻辑，当输入栏名字为空，或者输入栏密码为空时，游戏会提示当前的输入为空，提示玩家重新输入。



以下是isExit的内容：用于检验当前用户是否存在



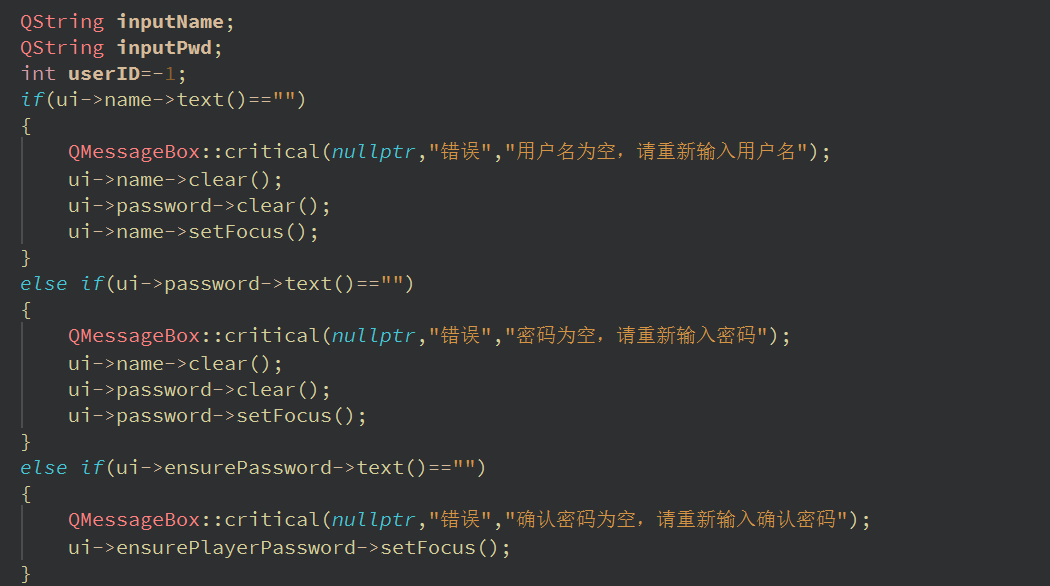
如果存在，则返回true，否则返回false，提示玩家重新登录游戏账号。

可以看到Login页面的右边有一个注册按钮，接下来介绍一下注册页面。

## 注册页面的实现



注册页面的实现思路也是基本一致：





这里玩家需要输入用户名，设置密码以及再次确认。如果三者中有一个为空，或者有设置密码和再次确认的密码内容不同时，游戏会提示玩家重新输入。如果全部内容都满足了，则游戏会实现同上面类似的操作，判断当前用户名是否存在，如果存在，则警告玩家不可以使用该用户名，请使用其他的用户名进行注册。

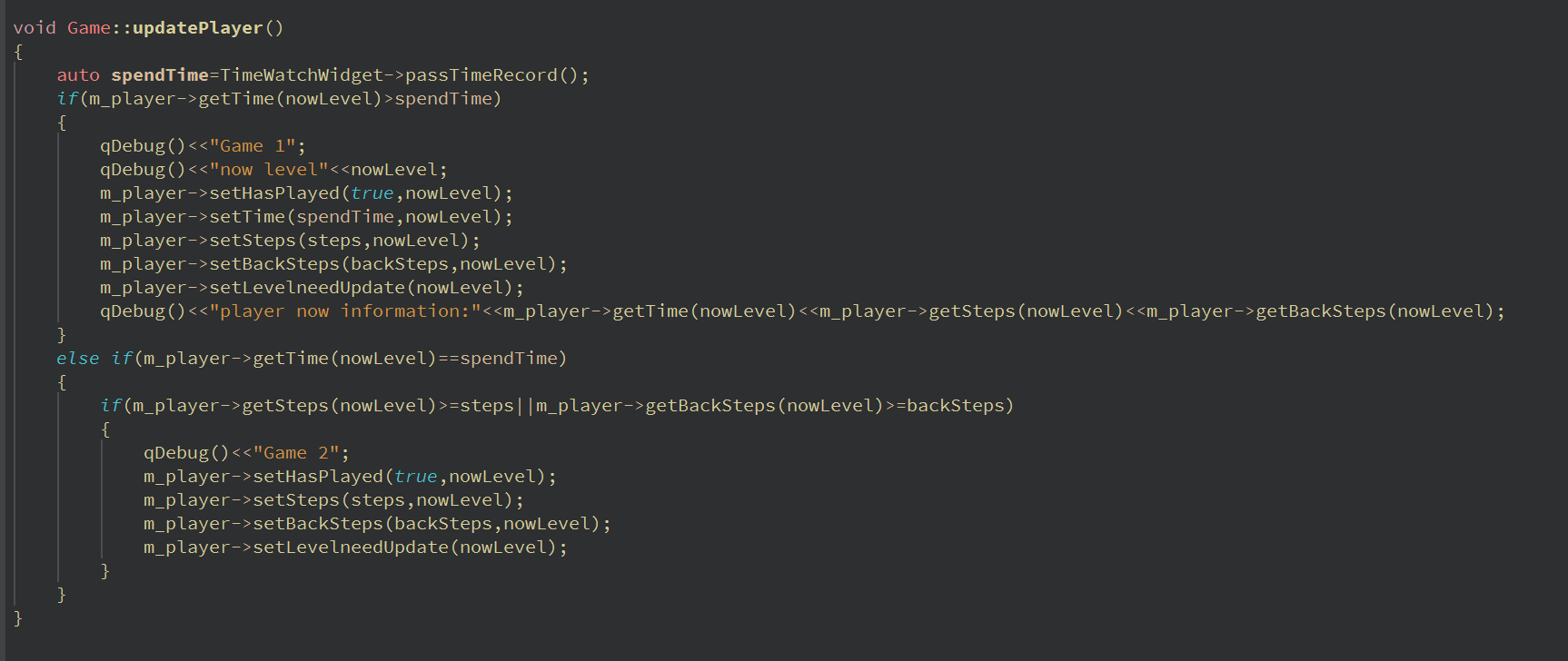
如果注册成功，则游戏会自动将新的游戏账号添加到文件中。

## 个人游戏账户页面的实现

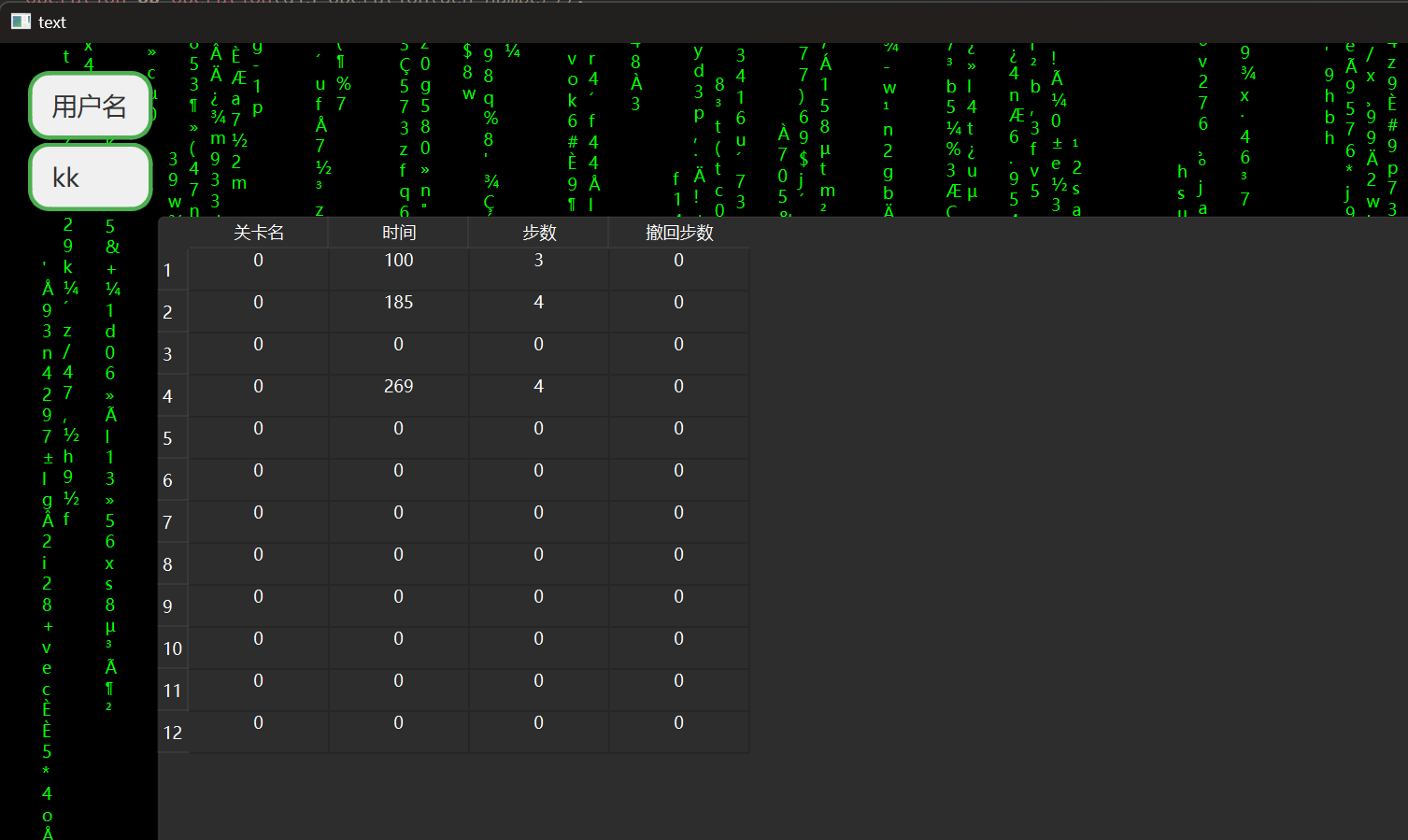


这是一个用户个人的游戏账户页面，玩家每通过一个关卡，就可以看到自己的这个关卡的最好游戏记录，包括操作步数、撤回步数、以及花费的时间。

当玩家登录完直接的账号以后，可以来到个人的游戏账户页面，可以看到自己之前的游戏记录。玩家可以玩完一把游戏后，再次返回个人游戏账户页面，此时可以发现相关的游戏记录也被更新了（更新的前提是游戏记录更好，这里以时间为判断依据）。



这里，m\_hasPlayed就排上用场了，后续打算添加一下音乐模块，丰富一下内容。



基本上本周的所有工作内容都在此完成了，目前其实正在设计排行榜，但是出了一些问题，正在寻求解决方法。