# 蓝顿蚂蚁第二次工作报告

软件1班 202430551743 严嘉鹏

**一、下载Qt并尝试在VS上部署**

原计划是先在VS上写好底层逻辑代码，最后通过提前设置好的接口来链接到Qt上实现可视化。然而，本周了解到Qt相对独立，学习路线陡峭。为了避免后期难以链接，修改计划为，一边跟Qt网课，一边写逻辑代码，到实训后期就可以逐渐添加Qt元件以实现GUI。

所以这周尝试在VS上部署Qt,通过镜像网站下载好Qt后,花费较长时间仍然卡在了编译器问题上，所以部署失败，决定先用Qt creator原生开发工具进行GUI学习。

**二、游戏主页面类的创建**

上次实验完成了自由模式的雏形，本次实验补上游戏主页的代码实现。添加了一些类和函数。运用了本周实验Muti-Class的学习成果，尝试各类之间互相调用函数和创建对象，效果良好。

类定义和函数原型代码如下：（由两个类构成）

#ifndef GAME\_HOMEPAGE\_H

#define GAME\_HOMEPAGE\_H

#include <iostream>

#include <string>

#include<fstream>

using namespace std;

class gameMode//游戏模式的种类

{

public:

int \_freedomMode();

void \_levelMode();

void \_funMode();

void \_competitionMode();

void \_cooperationMode();

};

class gameHomepage//主页面元素

{

public:

bool isUserRegistered(const std::string& username);//检查用户是否注册

void userReginster(const string& \_username, const string& \_password);//用户注册

void userLogin(const string& username1, const string& password1);//用户登录

void beginGame();//开始游戏

void chooseMode();//选择游戏模式

gameMode mode;//在gameHomepage类里定义gameMode类对象

};

#endif

相关函数定义代码如下：

void gameHomepage::beginGame()

{

cout << "欢迎来到兰顿蚂蚁游戏" << endl;

while (true)

{

cout << "请输入用户名" << endl;

string username;

cin >> username;

if (isUserRegistered(username))

{

cout << "用户已注册" << endl;

cout << "请输入密码" << endl;

string password;

cin >> password;

userLogin(username, password);

break;

}

else

{

cout << "用户未注册,请先完成账号注册" << endl;

cout << "请输入用户名" << endl;

string username2;

cin >> username2;

cout << "请输入密码" << endl;

string password2;

cin >> password2;

userReginster(username2, password2);

}

}

}

void gameHomepage::chooseMode()

{

cout << "请选择游戏模式" << endl;

cout << "1.自由模式" << endl;

cout << "2.闯关模式" << endl;

cout << "3.趣味模式" << endl;

cout << "4.对抗模式" << endl;

cout << "5.合作模式" << endl;

int choice=0;

cout << "请输入你想参加的模式对应的序号" << endl;

cin >> choice;

switch (choice)

{

case 1:

mode.\_freedomMode();

break;

case 2:

mode.\_levelMode();

break;

case 3:

mode.\_funMode();

break;

case 4:

mode.\_competitionMode();

break;

case 5:

mode.\_cooperationMode();

break;

}

}

**三、用户注册和登录**

由于本周刚好学了ofstream和ifstream文件输入输出流，前者以写访问的形式打开文件，后者以读访问的形式打开文件。本次实验利用不同的文件名来表示不同用户，实现了用户注册和用户登录。这种方式一旦记录，就会保存用户的账号名和密码，可以很好的和后续如何保存用户游戏信息以及制作排行榜相关联。由于时间有限，还有一些功能未实现，如用户忘记密码后的操作、如何隐藏用户输入的密码、如何删除用户记录，下次实验来完成。

函数定义的代码如下：

void gameHomepage::userReginster(const string& \_username,const string& \_password)

{

string filename = \_username + ".txt";

ofstream outFile(filename);//ofstream以写访问的形式打开文件

if (outFile.is\_open())//判断文件是否打开成功

{

outFile << \_password << endl;

cout << "注册成功" << endl;

outFile.close();

}

else

{

cout << "注册失败" << endl;

exit(1);

}

}

void gameHomepage::userLogin(const string&username1,const string& password1)

{

string filename = username1 + ".txt";

ifstream inFile(filename);//ifstream以读访问的形式打开文件

if (inFile.is\_open())

{

string storedPassword;

inFile >> storedPassword;

inFile.close();

if (storedPassword == password1)

{

cout << "登录成功！欢迎" << username1 <<"的到来！" << endl;

}

else

{

cout << "登录失败！密码错误！" << endl;

}

}

else

{

cout << "登录失败！用户不存在！" << endl;

}

}

bool gameHomepage::isUserRegistered(const std::string& username)

{

string filename = username + ".txt";

ifstream inFile(filename);//ifstream以读访问的形式打开文件

if (inFile.is\_open())//判断文件是否打开成功

{

inFile.close();

return true;

}

else

{

return false;

}

}

**四、实验心得**

本周很幸运，学到了多类和文件流，刚好可以运用到实训里。在实验中也是遇到很多困难，比如，类定义时要考虑好各类的顺序否则会报错、类和对象互相参杂很容易出现没有正确连接和调用的问题。遇到问题要有耐心，没思路时可以先完成个大概再求助大模型进行删减。另外，国内下载某些计算机软件太慢了，其配置难度也较高，需要花费较长时间。其他：由于暂时尚未学会利用Git托管代码，所以目前只在GitHub上传了工作报告和源码复制。

**五、运行成果**

注册了两名用户和相关密码，以下为他们的游戏过程。

Jack 123456

Mike 12345678

