Assessment 3

Rock Paper Scissors simulation version 2.0

Problem statement

The program should be able to have a complete main menu interface, including: login and registration interface, logout page, game interface. The system should be able to record the user's game data and be able to retrieve and clear it at any time.

1. Make an interface to let the user chose login-in or registration (Subroutine start ()).

2. Login-in & registration interface (subroutine Login () & Register ()).

3. Establish a menu interface, let the user chose which one function (1. play the game; 2. view the history of the game; 3. clean the data of the user and 4. login-out the program) they want to use.

4. It will back to the menu interface to quit the program.

Analysis

The most important part of this issue is program need to create a file to save the user data by itself and program need locate the data from the file precisely. In one word, how to save and out a file is the point.

1. Input:

User need choose to login (or registration, if it does not have an account) the system. The user should enter the username and password, the program will compare the information with base data, if all the character are the same, the system will jump to the next menu, conversely, if the information has one difference, the system will ask the user reenter the information again.

1. Output:

When account login successfully, the program should clean the screen and enter the menu subroutine and wait the user select the options. When user output the “history data”, the data of this user will print on the screen. In the part of the game subroutine, the program will print the phase of the game ou the screen.

1. Data structure:

This issue only involves with number and characters, so data structure is int. and char..

1. Algorithm:

The variables which need to deal with are all number, so, the algorithm is only basic math.

Design:

Login part:

1. Make an interface to hint the user enter the username and password by used scanf();.
2. Check the data has saved in the local file or not, if local file has not this data, turn to the subroutine register().
3. Open the data file by fp=fopen(); and get the password from user again, compare the password with the local data, turn to the menu interface and wait for next step if there is no difference between the data.
4. If the password is wrong, system should ask the user would like to reenter the password again or back to the subroutine register() again.

Registration part:

1. Use the scanf(); enter the username and password from user.
2. Open the data file by fp=fopen(); to get the user data from the structure, to ensure this username is the first time to register in this system.
3. Write the information into the data file and close the file.
4. Turn to the subroutine Login(); to continue the system.

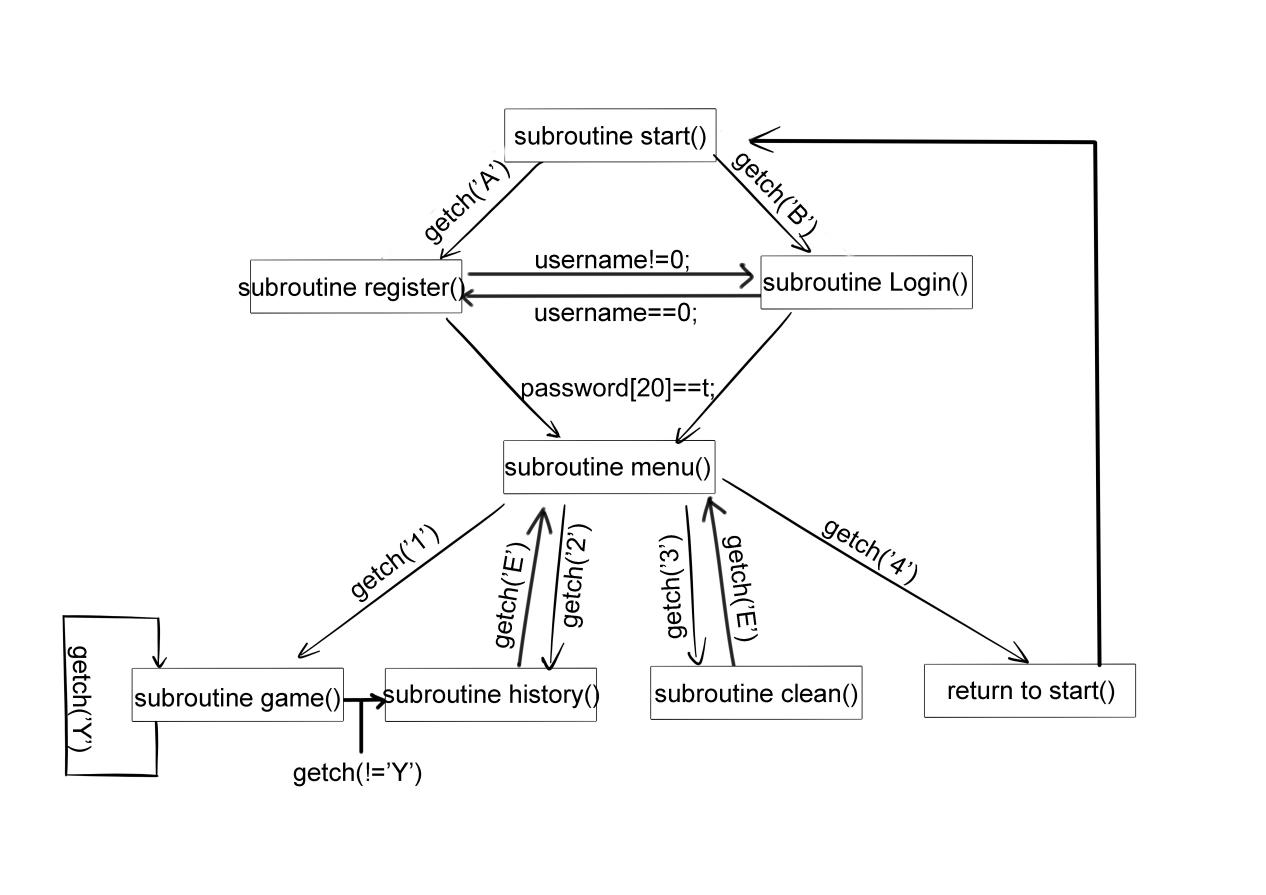
Menu part:

Use the if loop and getch(); let the system have function to: start a new game; review the history game data; clean the game data and quit the account.

Game part:

Use the assessment 2 C program code.

Flow chart:



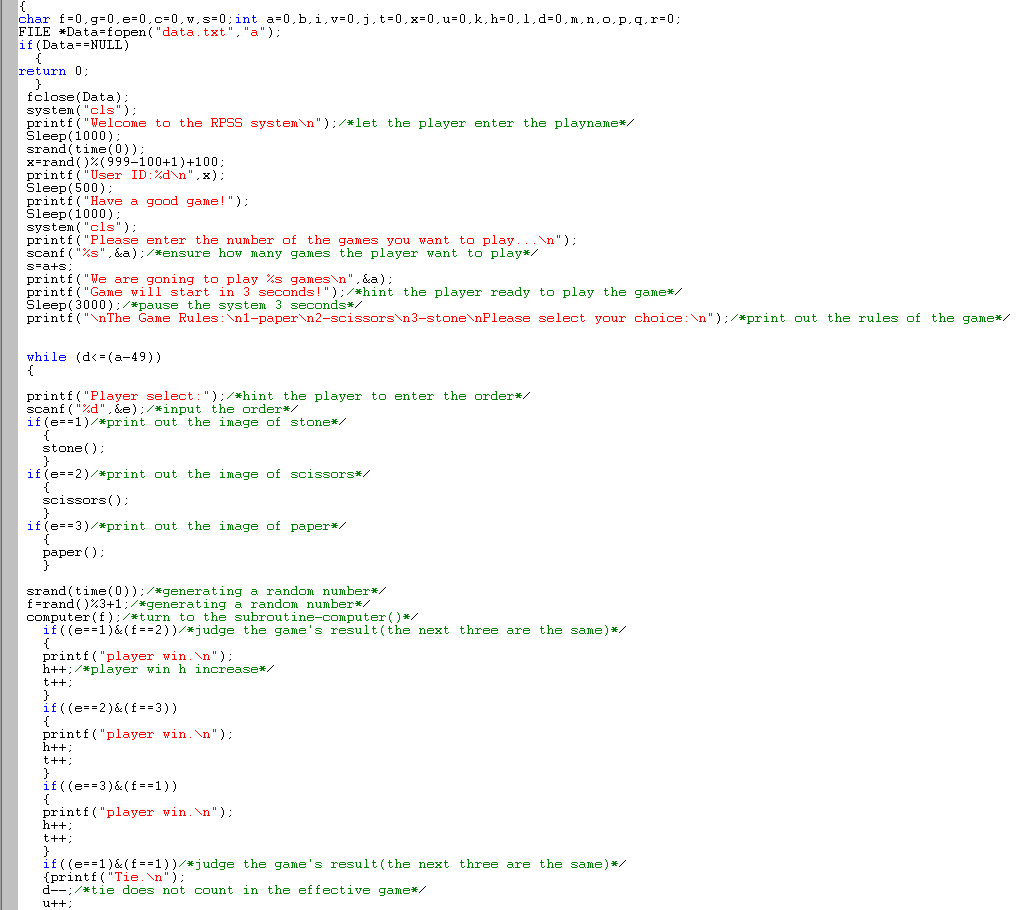
Implementation

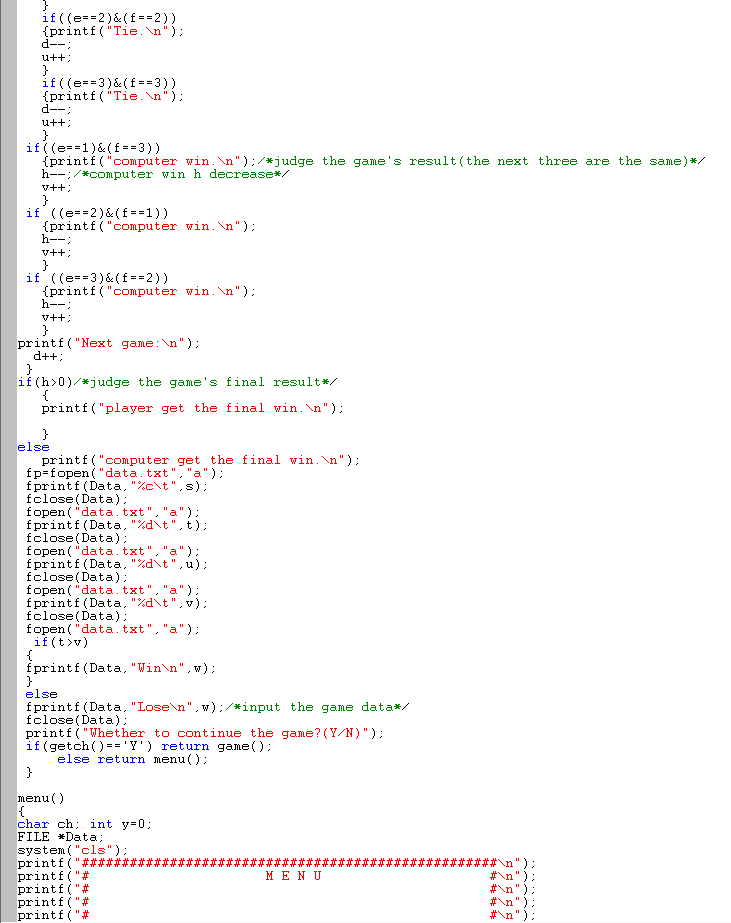
The code is as follows:

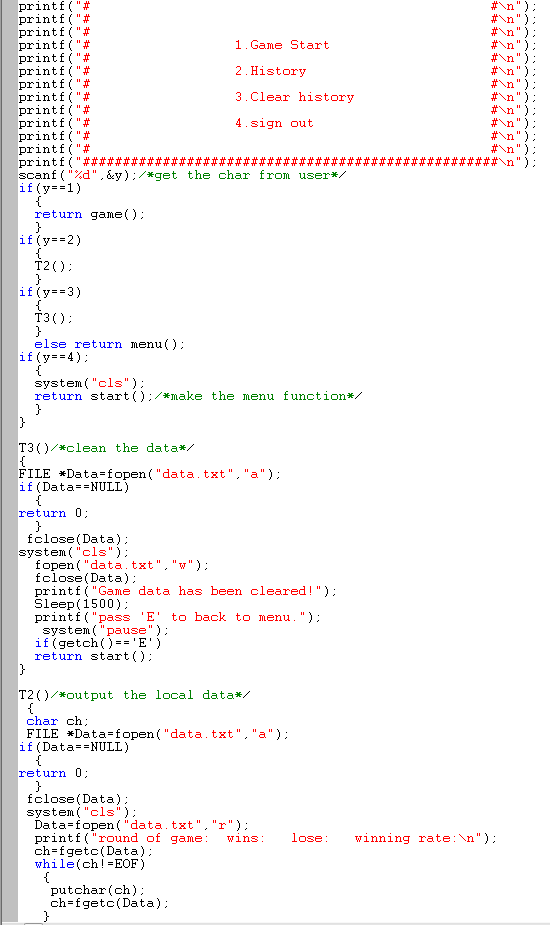


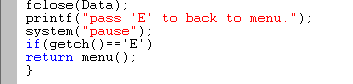




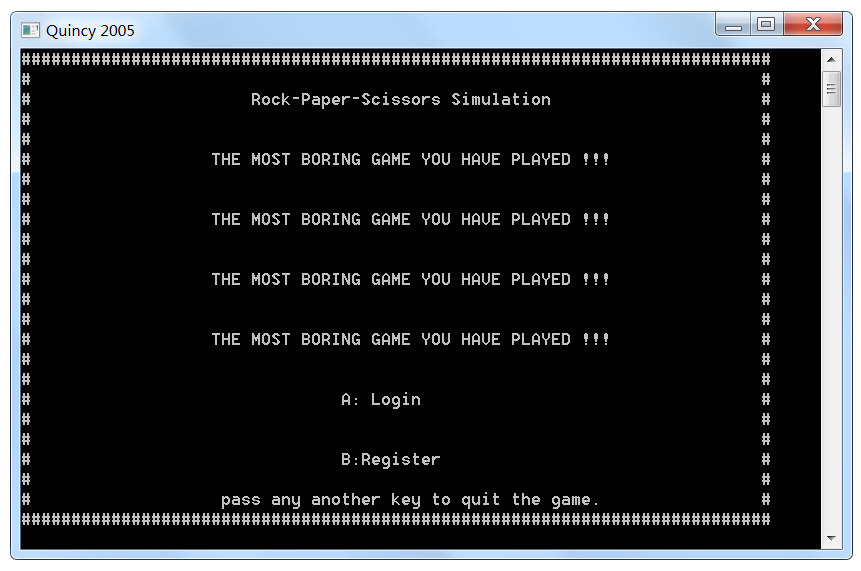




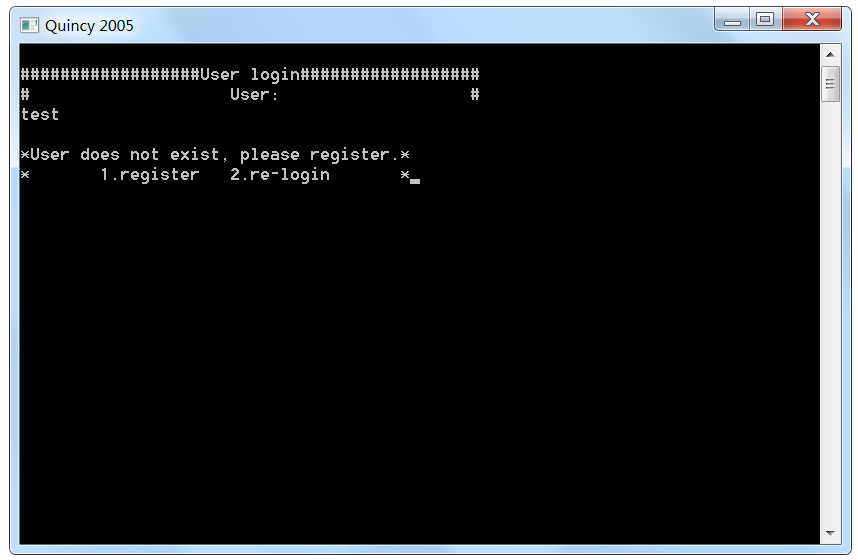




Testing:

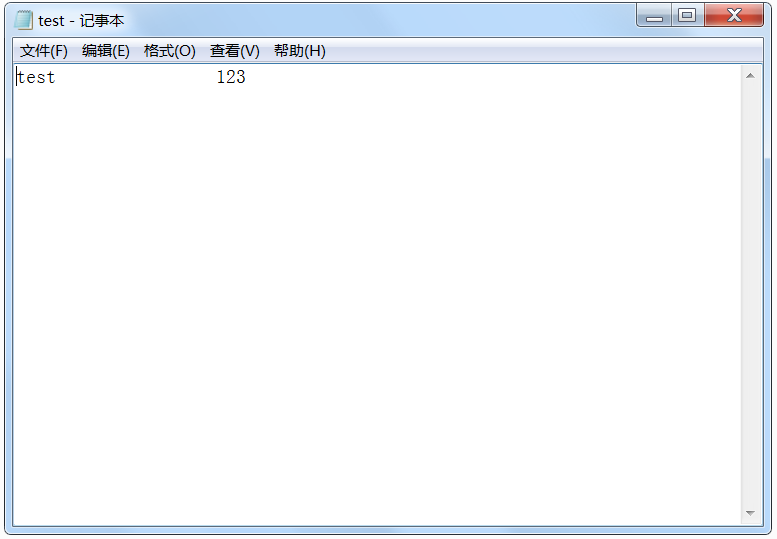


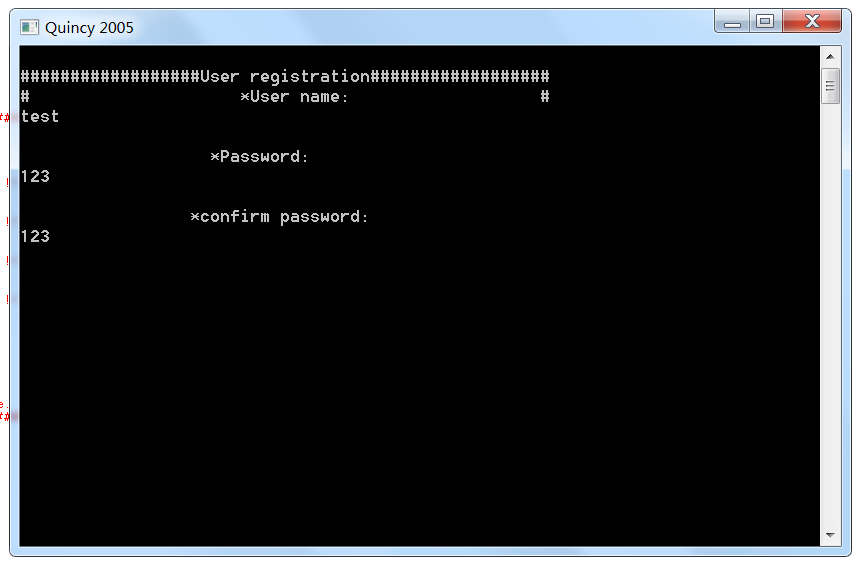
The program can be started normally, and it has a Design interface, enter the ‘A’ and enter a nonexistent username, in order to know what will the program do when user enters an illegal name.

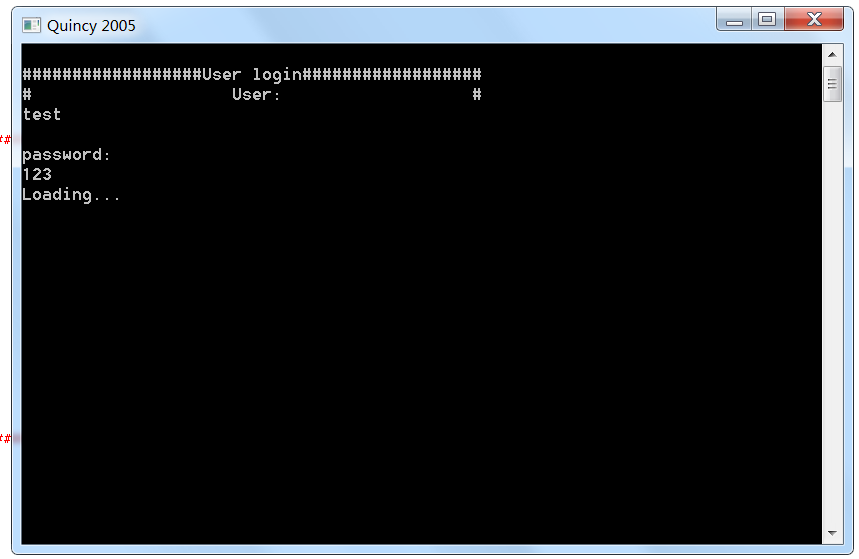


The program finds the issue and request user to reenter the data or register a new account. After registered the username, the system creates an account file, which saves the username and password in, and turn to the subroutine Login(); to continue the program. (The record of the phase of registration will be cleaned by program when it finished, so this part has no screenshot.)

t3



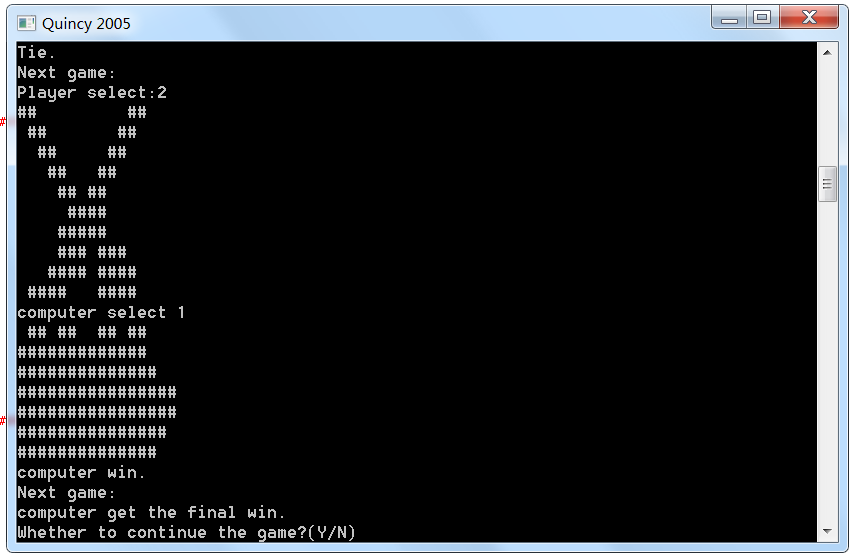
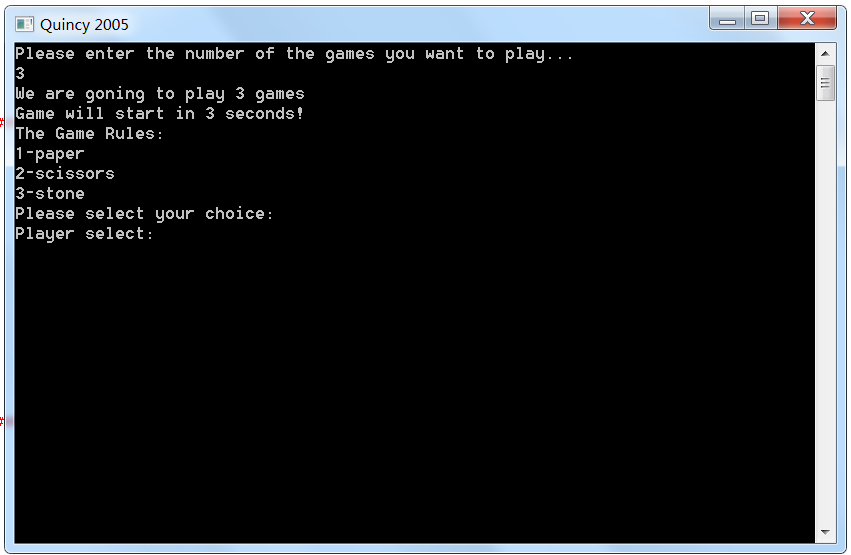
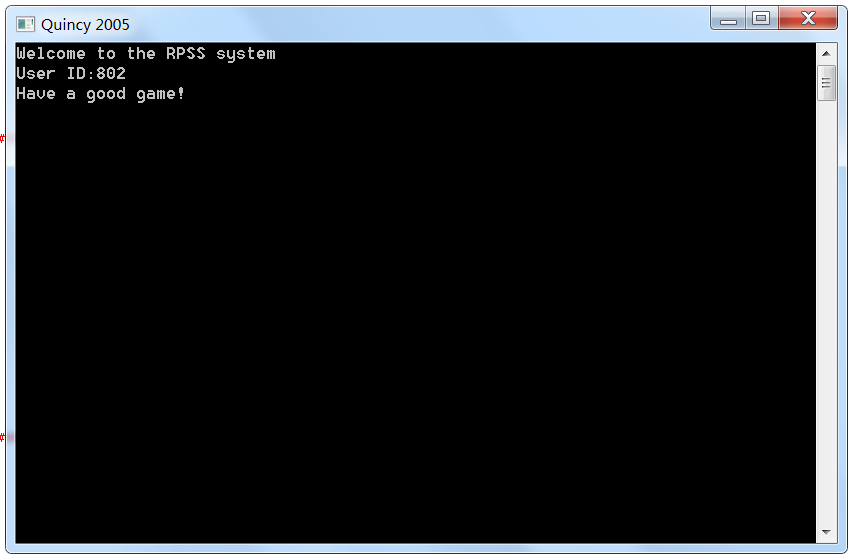




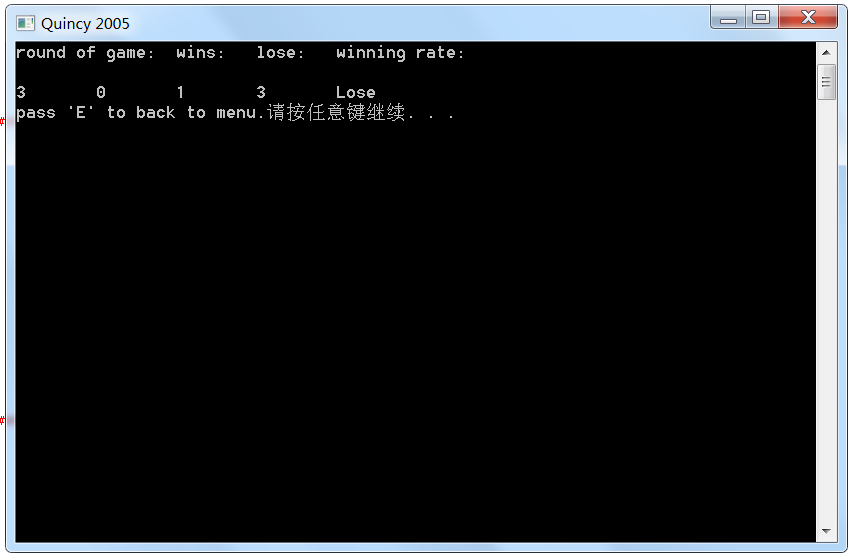
After registered, enter the account information, login in the system successfully.



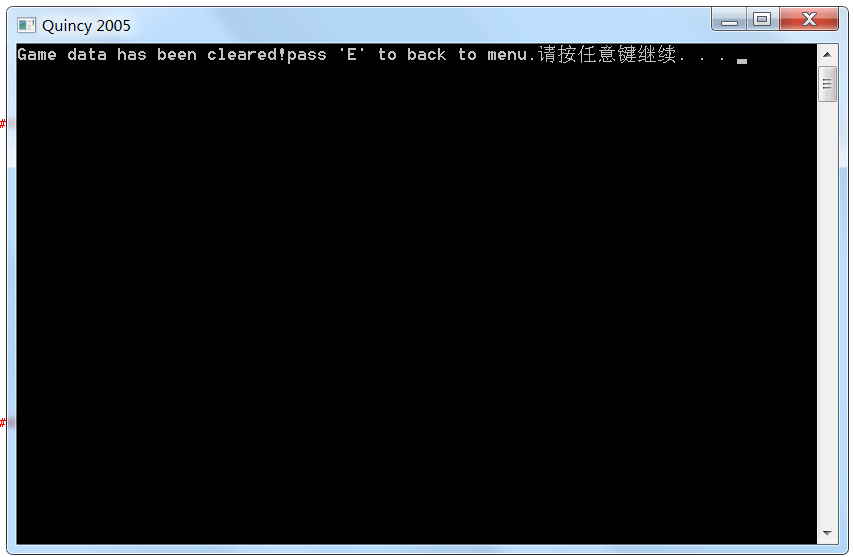
Enter the menu interface, and enter the ‘1’ start the game.



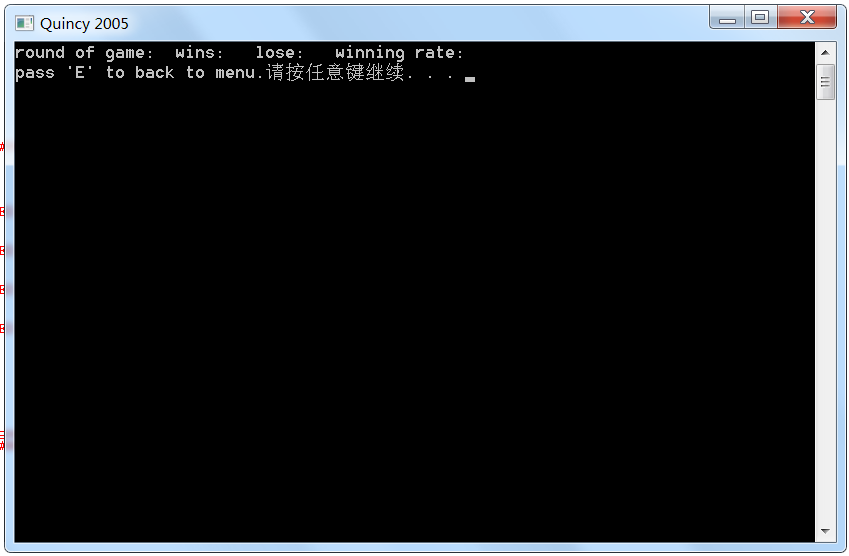
After the game, user can enter the ‘y’ to continue the game or back to the menu interface. Then enter the ‘2’ to print out the game data out on the screen.



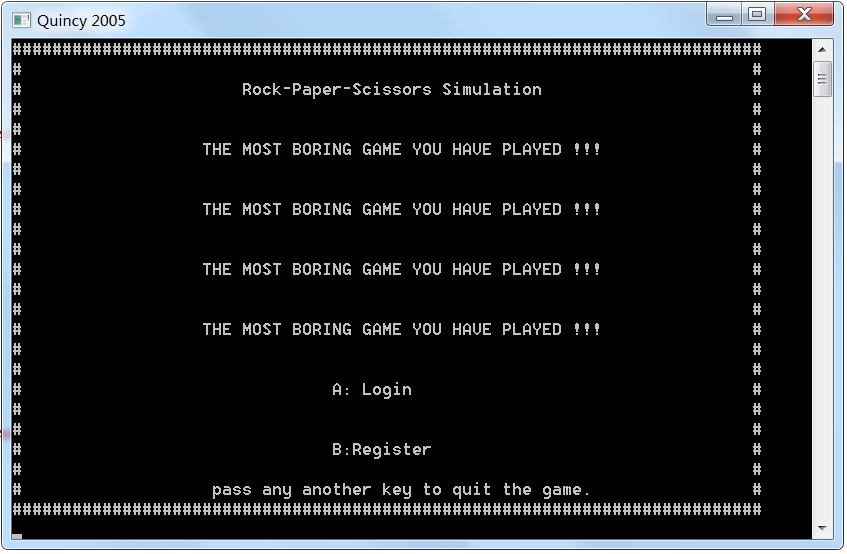
Enter ‘3’ to clean the personal data.



The data of the user has cleaned up successfully.



Enter ‘4’ to log out the account, and back to the start interface.



The program has finished successfully, and no obvious bug. This is a kind of complete design, but still has some area that can be improved.