



可视化程序设计

翔乐游戏平台

班 级： 21534

学 号： 2153412

姓 名： 石鹏翔

指导教师： 沈哲

东北大学秦皇岛分校计算机与通信工程学院

2018.01

目录

1、系统简介

- 1.1 功能（任务描述）
- 1.2 功能需求分析
- 1.3 用户使用流程
- 1.4 软件环境

2、模块设计

- 2.1 整体设计（成员分工分数分配）
 - 2.1.1 完整游戏平台设计
 - 2.1.2 本平台的设计
 - 2.1.3 工作分配
 - 2.1.4 用户信息类：CustomInfo
 - 2.1.5 验证码的生成类：VerifyCodeImgSimple
- 2.2 主界面设计
 - 2.2.1 未登录主界面
 - 2.2.2 主界面（单机游戏）
 - 2.2.3 主界面（网页游戏）
 - 2.2.4 用户信息管理界面
 - 2.2.5 平台信息界面
- 2.3 登录，注册界面设计
 - 2.3.1 登录界面
 - 2.3.2 注册界面
- 2.4 游戏设计
 - 2.4.1 贪吃蛇（主要我实现的）
 - 2.4.2 连连看（主要我实现的）

3、数据库设计

4、开发心得

翔乐游戏平台

1.系统简介

1.1 功能：

本程序是一个游戏平台，提供各种小型单机游戏，以及一些网页游戏。对于用户来说，我们提供了用户信息管理系统。在这个平台下我们有 12 个小型单机游戏，对于各个游戏，我们有相应的得分系统，以及排行榜。在这个平台下还有一些网页游戏。网页游戏依靠第三方浏览器平台。

1.2 功能需求分析

在当今生活中，游戏是一个放松身心的重要途径，也是不可或缺的一部分。适度的玩游戏能够帮助我们缓解疲劳，休闲娱乐。所以本次大作业，我们做了一个游戏平台。本平台集成一众多小型游戏，界面优美，功能健全。

1.3 用户使用流程

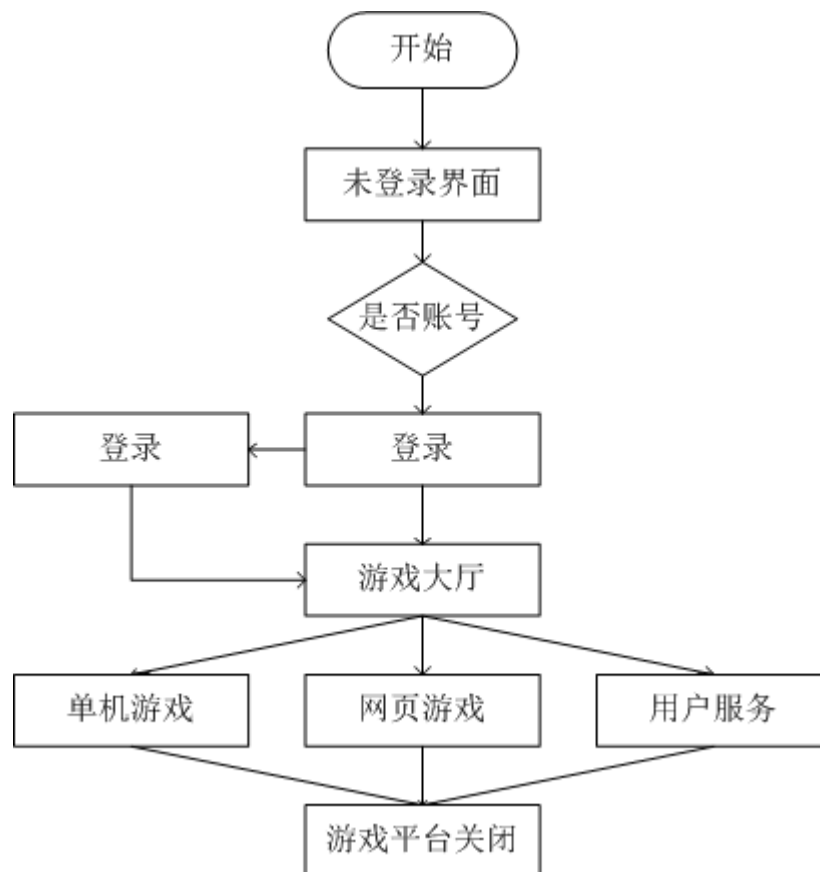


图 1 用户使用流程图

1.4 软件环境

利用 Visual Studio 2015 作为前台开发工具，后台 SQL server2016 实现数据库系统设计，PS 做美工设计。

2.模块设计

2.1 整体设计

2.1.1 完整游戏平台设计

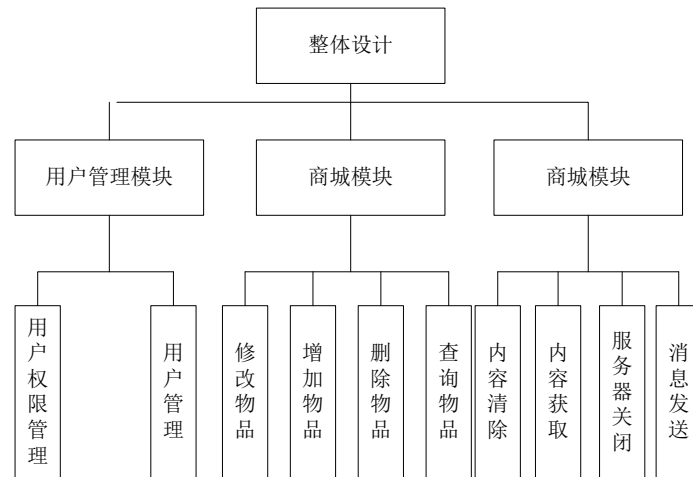


图2 游戏平台完整设计模块图

这个是一般游戏平台的完整设计，但是实现的过程中我们发现这个实现有些困难，C#网络编程不是很熟练，有些模块涉及其他的技术，需要其他知识的支持，由于缺乏这些知识，所以对游戏平台的功能进行了删减和转移。我们的重心放在了界面的设计优化和平台游戏的开发还有平台的管理上。

2.1.2 本平台的设计

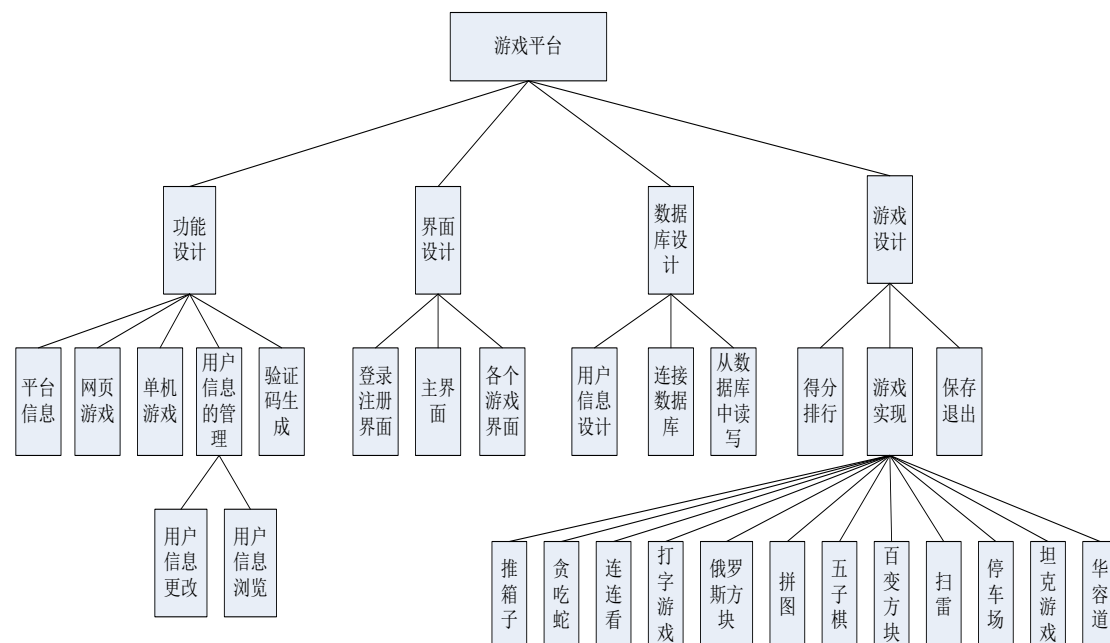


图3 本平台的设计模块图

2.1.3 工作分配

此游戏平台的开发主要工作有：功能设计，界面设计，数据库设计，游戏设计。工作的分配如下：

表1 成员任务分配表

学号	成员	工作	分数分配(10分)
2153412	石鹏翔	功能设计, 界面设计, 游戏设计(贪吃蛇, 连连看)	5
2153526	王云	数据库设计, 游戏设计(五子棋, 推箱子, 华容道, 百变方块, 扫雷)	4
2153406	于靖权	游戏设计(俄罗斯方块, 打字游戏, 停车场, 拼图, 坦克大战)	1

2.1.4 用户信息类: CustomInfo

对玩家进行封装, 主要有如下属性和方法

表2 CustomInfo类的成员变量表

成员变量	作用
cname	玩家昵称
sex	玩家性别
email	玩家账号
passwd	玩家密码
pic_num	玩家头像序号 (男、女玩家各有四种头像)
grade	玩家等级
money	玩家的金币数目

表3 CustomInfo类的成员函数表

成员函数	作用
GetSQLData	从数据库中得到用户的信息
Update	更新用户的信息
InsertData	向数据库中写入玩家信息

2.1.5 验证码的生成类: VerifyCodeImgSimple

验证码图片的生成操作的封装, 主要有如下方法

表4 VerifyCodeImgSimple类的成员函数表

成员函数	作用
CreateVerifyCode	生成随机字符串
CreateImage	生成校验图片

2.2 主界面设计

主界面上需要显示玩家的信息, 单机小游戏的显示, 还有平台的信息的显示, 还要兼顾平台的美观、便捷。

2.2.1 未登录主界面：



图 4 未登录主界面

2.2.2 主界面（单机游戏）：



图 5 主界面（单机游戏）

2.2.3 主界面（网页游戏）：



图 6 主界面（网页游戏）

2.2.4 用户信息管理界面：



图 7 用户信息管理界面

2.2.5 平台信息界面：



图 8 平台信息界面

2.3 登录注册界面

登录注册界面的设计主要是关于向数据库中写数据，还有从数据库中读数据。

2.3.1 登录界面：



图 9 登录界面

2.3.2 注册界面:



图 10 注册界面

2.4 游戏设计

2.4.1 贪吃蛇

该部分有实现的难点有：蛇的移动，以及吃到水果的判断，游戏失败的判断，还有游戏界面的设计。主要有如下几个部分：

2.4.1.1 食物类 Bean

主要有两个属性，foodtype 为食物类型（苹果和梨子），origin 定义位置字段。还包括显示食物的方法 Display 和消除食物的方法 UnDisplay。

2.4.1.2 块类 Block

该类是界面中每一个点的实现，主要包括 3 个成员变量，number 定义序数字段，origin 定义位置字段，ishead 定义蛇头标志字段。还有两个成员函数，一个 Display 为显示块的方法，UnDisplay 为消除块的方法。

2.4.1.3 界面类 Floor

该类是为了界面的实现，主要有 8 个成员变量，foodtype 为食物类型，unit 定义静态单位长度，length 为运动场界面的长度，width 为运动场的宽度，dot 为运动场左上角的位置，score 为游戏得分，s 为蛇对象，bean1 为食物对象。还有一些方法，显示新食物的方法 displaybean，重新设置蛇方法 ReSet，产生随机食物的方法 randombean，显示运动场界面的方法 Display，其参数为一图形对象，检查食物是否被吃的方法 CheckBean，其参数为图形对象，检查蛇是否撞墙的方法 CheckSnack。

2.4.1.4 蛇类 Snake

该类定义一个蛇类，主要包括如下属性，数组字段 blockList，蛇头序数或蛇长度 headNumber，蛇头位置 headPoint，方向 direction。还有一些成员函数，Snake 为构造函数，判断蛇是否碰到自身的方法 getHitSelf，蛇的转向方法 TurnDirection，其参数为参数为蛇要改变方向，蛇增长方法 Growth，蛇前行的方法 Go，显示蛇的方法 Display，消除蛇的方法 UnDisplay，重新开始游戏的方法 Reset。

2.4.1.5 游戏界面



图 11 贪吃蛇游戏界面

2.4.1.6 排行榜



图 12 贪吃蛇游戏排行榜界面

2.4.1.6 帮助界面

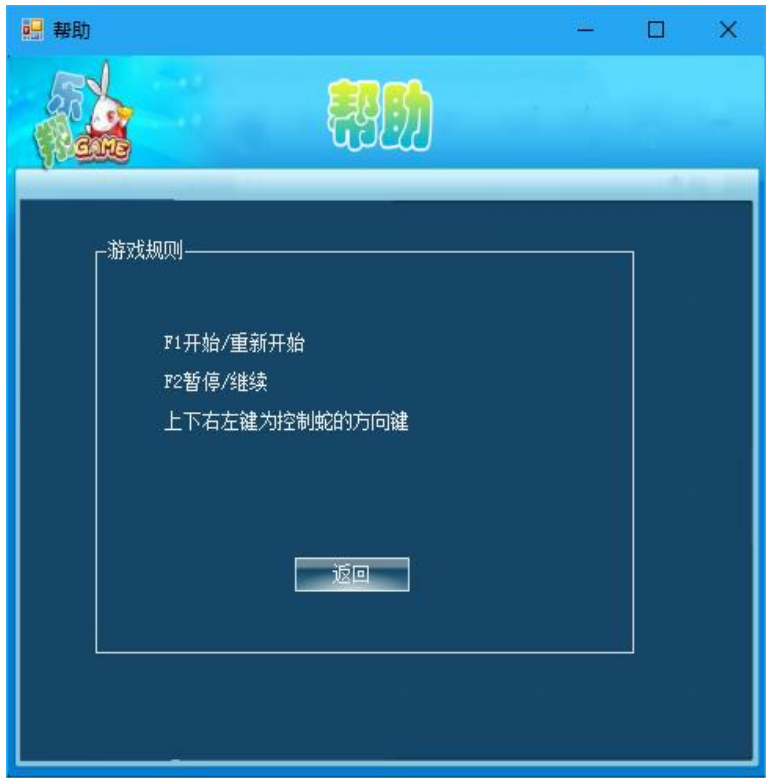


图 13 贪吃蛇游戏帮助界面

2.4.2 连连看

该游戏的设计难点在于判断游戏的整体设计，和判断两方块是否相同且联通，还有游戏界面的设计。主要有如下几部分：

2.4.2.1 窗体 Lian_F

该窗体的成员变量和成员函数如下表所示：

表 5 Lian_F 的成员变量表

成员函数	作用
m_map	游戏地图
GameSize	布局大小即行列数
LType	联通方式
Select_first	是否已经选中第一块
Source0	所有动物的图片
W	动物方块图案的宽度
x1	被选中第一块的地图坐标
x2	被选中第二块的地图坐标
y1	被选中第一块的地图坐标
y2	被选中第二块的地图坐标
z1, z2	折点棋盘坐标
cemail	用户的账号
game_name	该游戏的名称

表 6 Lian_F 的成员函数表

Lian_F	构造函数
Form1_Load	页面加载函数
StartNewGame	开始新游戏函数
Init_Graphic	初始化界面函数
get_Graphic	生成 Graphics 对象
create_image	实现按标号 n 从所有动物图案的图片中截图
IsWin	检测是否已经赢得了游戏
IsSame	判断是否相同
X_Link	X 直接连通即垂直方向连通
Y_Link	Y 直接连通即水平方向连通
OneCornerLink	1 直角接口连通
TwoCornerLink	2 折点连通
XThrough	水平方向判断到边界的连通性
YThrough	垂直方向判断到边界的连通性
IsLink	判断选中的两个方块是否可以消除
Delay	让程序延时数秒
DrawLinkLine	画选中方块之间连接线
UnDrawLinkLine	清除选中方块之间连接线
DrawSelectedBlock	画选中方块的示意边框线
ClearSelectedBlock	清除方块
保存得分 ToolStripMenuItem_Click	向数据库中写数据，保存得分
退出 ToolStripMenuItem_Click	退出
排行榜 ToolStripMenuItem_Click	打开排行榜窗体
帮助 ToolStripMenuItem_Click	打开帮助窗体
开始游戏 ToolStripMenuItem_Click	开始游戏

2.4.2.2 判断是否连通的流程图

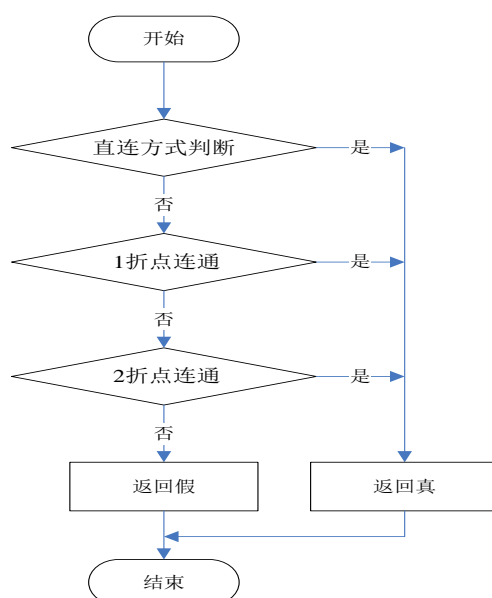


图 14 判断是否连通流程图

2.4.2.3 整体设计

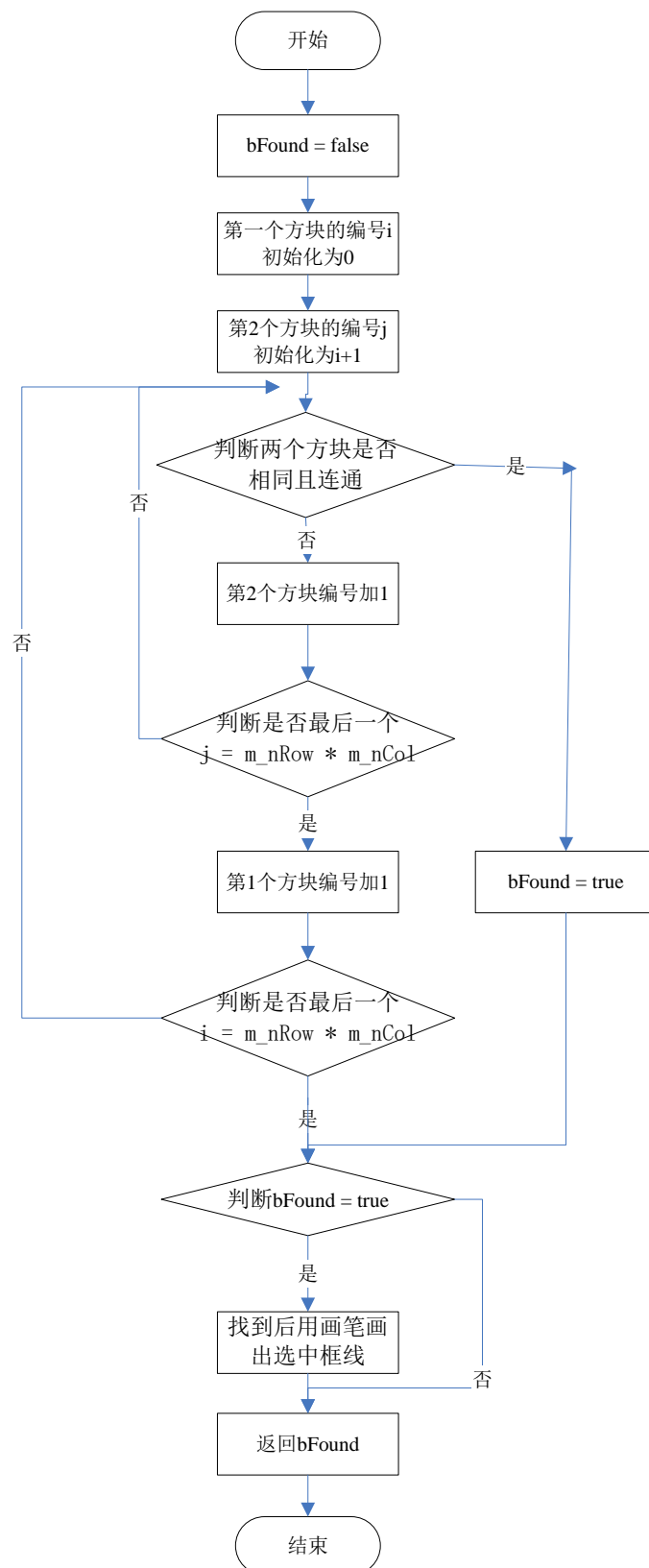


图 15 游戏流程图

2.4.2.4 游戏界面



图 16 连连看游戏界面图片

2.4.2.5 排行榜和帮助界面

界面与贪吃蛇类似，不再赘述。

3、数据库设计

3.1 玩家信息记录

3.1.1 属性介绍

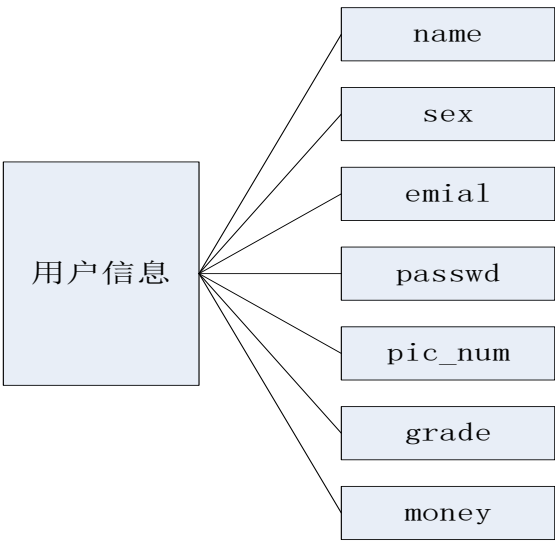


图 17 玩家信息属性图

	列名	数据类型	允许 Null 值
▶	name	varchar(50)	<input type="checkbox"/>
	sex	char(2)	<input type="checkbox"/>
🔑	email	varchar(50)	<input type="checkbox"/>
	passwd	varchar(50)	<input type="checkbox"/>
	pic_num	int	<input type="checkbox"/>
	grade	int	<input checked="" type="checkbox"/>
	money	int	<input checked="" type="checkbox"/>

图 18 数据库中玩家信息表

3.1.2 数据库中实际值

	name	sex	email	passwd	pic num	grade	money
▶	石鹏翔	男	123	123	3	31	0
	石鹏翔	男	aa	aa	2	100	10
	ww	女	aq	as	1	50	10
	曹萌	男	dd	dd	2	2	0
	spx	男	spx	spx	3	20	12
	ww	男	ww	ww	1	202	1000
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

图 19 数据库中实际值

3.2 各个游戏玩家的得分表

3.2.1 属性介绍

	列名	数据类型	允许 Null 值
🔑	email	varchar(50)	<input type="checkbox"/>
	score	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

图 20 数据库中游戏信息表

3.2.2 数据库中的值

	email	score
▶	dd	101
	ww	1
*	NULL	NULL

图 21 数据库中实际值

3.3 平台信息表

3.3.1 属性介绍

	列名	数据类型	允许 Null 值
▶	plat_owner	varchar(50)	<input type="checkbox"/>
	version	varchar(50)	<input type="checkbox"/>
🔑	tell	varchar(50)	<input type="checkbox"/>
	company	varchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

图 22 数据库中平台信息表

3.3.2 数据库中的值

	plat owner	version	tell	company
▶	翔、王云、于靖权	1.1.0	18533570730	翔乐游戏公司
*	NULL	NULL	NULL	NULL

图 23 数据库中实际值

3.4 玩家充值记录

3.4.1 属性介绍

	列名	数据类型	允许 Null 值
🔑	email	varchar(50)	<input type="checkbox"/>
	money	int	<input type="checkbox"/>
▶	time	time(7)	<input type="checkbox"/>
			<input type="checkbox"/>

图 22 数据库中玩家充值记录表

3.4.2 数据库中的值

	email	money	time
	ww	10	12:30:10
	123	100	12:30:10
▶*	NULL	NULL	NULL

图 23 数据库中实际值

3.5 从数据库中读数据的类 Data_Game_R

```
using System;  
using System.Collections.Generic;  
using System.Data.SqlClient;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```

namespace GamePlatform.Con_code
{
    class Data_Game_R
    {
        string email;        //用户账号
        int score;           //得分
        public string Email
        {
            get { return email; }
            set { email = value; }
        }
        public int Score
        {
            get { return score; }
            set { score = value; }
        }

        public void GetSqlData(string tab_name, string Sql_email)
        {
            SqlConnection conn = new
SqlConnection(GamePlatform.register.connectionString); //新建数据库连接
            conn.Open(); //打开连接
            string Inquire = "select email,score from "+tab_name+" where email=' " +
Sql_email + "' "; //查询语句
            SqlCommand cmd = conn.CreateCommand();
            cmd.CommandText = Inquire;
            SqlDataReader reader = cmd.ExecuteReader();
            while (reader.Read()) //获取查询结果
            {
                email = reader.GetString(reader.GetOrdinal("email"));
                score = reader.GetInt32(reader.GetOrdinal("score"));
            }
        }
    }
}

```

3.6 向数据库中写数据的类 Data_Game_W

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;

```



```

using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace GamePlatform.Con_code
{
    class Data_Game_W
    {
        SqlConnection conn = new SqlConnection(GamePlatform.register.connectionString);
        public void Insert_Data(string tab_name, Data_Game_R Insert_data)
        {
            //查询是否存在数据
            //若不存在，则加进去，
            //若存在，比较大小，
            //若现在的数据大于数据库中的数据，则插入数据库
            //否则，不进行修改

            conn.Open();
            SqlCommand checkCmd = conn.CreateCommand();
            string s = "select email from " + tab_name + " where email=' " + Insert_data.Email
+ " " ";

            checkCmd.CommandText = s;
            //实例化数据适配器
            SqlDataAdapter check = new SqlDataAdapter();
            check.SelectCommand = checkCmd;
            DataSet checkData = new DataSet();
            int n = check.Fill(checkData);
            if (n != 0)
            {
                Data_Game_R old_data = new Data_Game_R();
                old_data.GetSqlData(tab_name, Insert_data.Email);
                if (old_data.Score < Insert_data.Score)
                {
                    //更新
                    SqlConnection MyConnection = new
SqlConnection(GamePlatform.register.connectionString);
                    string update = "Update " + tab_name + " set score=' " + Insert_data.Score
+ " " where email=' " + Insert_data.Email + " " ;//+"(email,score) values(' " + Insert_data.Email
                    SqlCommand mycom = new SqlCommand(update, MyConnection);
                    try
                    {
                        MyConnection.Open();

```

```

        mycom.ExecuteNonQuery();
        MyConnection.Close();
    }
    catch (Exception ex)
    {
        Console.WriteLine("{0} Exception caught.", ex);
    }
    //mycom.ExecuteNonQuery();
    conn.Close();

    mycom = null;
    conn.Dispose();
}
}
else
{
    //插入
    string insert = "Insert into " + tab_name + "(email,score) values(' " +
Insert_data.Email + "',' " + Insert_data.Score + "')";
    SqlCommand mycom = new SqlCommand(insert, conn);
    mycom.ExecuteNonQuery();
    conn.Close();

    mycom = null;
    conn.Dispose();
}
}
}
}

```

4.开发心得

我首先谈谈我自身的情况吧。在大二的时候我自己学过一点点 Android 开发方面的知识，也学了一点点 JAVA 语言，算是有一点点基础。在上老师讲的课的时候，实际操作方面，我都能听得懂，对于有些陌生概念的理解不是很深入，比如说接口、委托之类的。其实这个我以前看 Java 的时候就不是很懂，不是很理解，上完 C#课后，我对这些类似的概念有了更深入的了解，但是还是自己解释不清楚。但是在做完老师布置的这次作业后，我觉得我现在能解释

很多我以前不是很理解的知识点。我觉得我在这次作业感触还是很多的，所以我想把它写下来。

从拿到作业题目开始起，我就一直在想我到底做什么题目比较好，我该怎么去实现这个题目的要求。我不想随便在网上找找，看看哪个有代码就写哪个。我想找一个我想做的题目和我自己能力能做的题目。前前后后，断断续续的大概想了一个星期，我也没有想到我想去做的题目和适合我自己的题目。后来在一次玩 QQ 游戏的过程中，我就想我何不做一个单机游戏平台呢。参考了老师您的意见后，我就决定做一个游戏平台了。后来在开发的过程中，让我不断走下去，让我不断去优化的动力中，兴趣占了很大一部分。

大概 12.20 号考完试，我就开始了我的作业，开始的时候，虽然有想法，但是毕竟第一次上手，还是不知道怎么去实现，数据库也是一年以前学的，基本上都忘了，所以自己还是很迷茫的，不知道该怎么去写，就自己坐在那干想，想了一天，也没什么大的成果，vs 总是连不上数据库，总是出现报错，就拖出来了一个丑陋的界面，见图 24 未优化的图。恰巧有一个同学让我帮他找找代码，调一调，在这个过程中，我觉得我至少了解了 C#和数据库怎么连接，怎么从 VS 向数据库写数据，怎么从数据库读数据，怎么更新数据库中的数据。我就能很感觉出来，我知道了我该怎么去做了，第二天，我有花了一上午去看别人的程序，第二天晚上，我觉完成了用户登录的功能，万事开头难，我觉得那一天我就成功了一般。

经过 6、7 天的艰苦奋斗，平台大致功能已经实现了，看着自己的作品，心里还是很高兴的。我以为这样就差不多完事了，就差一个报告了。但是事实证明我想多了。

大概是 12 月 31 号，就想不 C#报告给写了，报告写了一半觉得自己这个平台有些 low 了，于是想着美化一下，反正也花不了多长时间，事实证明我想错了，这个优化很费时间，我原来觉得我做的不错，但是越优化越觉得自己做的不过好，什么地方都不好看，什么地方都得改，这一改有是 3 天时间，最终效果我觉得还是比以前好看很多，图 25 优化后的图。

总结：

1.有些事可能有些强制性，但是我们仍然能够找到我们的兴趣点，带着兴趣去做自己必须要做的事。毕竟兴趣是最好的老师。最后看到自己的成果，还是很有成就感的。

2.多看多写，看一下别人写的代码，会养成一个好的编程习惯。看别人的代码能开阔自己的思路。

2.多编多敲，看了代码，就要写代码，哪怕是和别人相同。特别是在编译的时候，能发现自己的进步和不足。在失败中取得微小的成功，是一件很快乐的事情。

3.简单编程，不要一开始就是很困难、好多行的代码。这样自己的自信心会被削弱的。自尊会被打击的。可以从最基本的开始做起，从一开始就慢慢体会代码重复利用的快乐。毕竟这样很节省代码的输入量，而且能体现到学习的效果。

4.多多交流。在本次作业中，一个同学告诉我 winform 优化界面 CSkin 比较好用。CSkin 节省了我很多时间，也使我的界面更加美化。

5.发现新思路，编程最麻烦的就是套用老方式和老办法。现实生活的游戏平台有很多，我们实现不了全部的，我们可以做一些创新，做单机游戏平台。创新是编程最大的灵感。

6.创造完美，有时候，自己觉得自己做的可以了，其实不然，只有自己深入进去，就会发现自己还有很多没有做好。

7.学会合作，只有通过合作才能创造完美。通过团队的努力奋斗，会发现自己的工作量减少了，自己的质量提高了，自己的效率提升了，自己的能力增强了。毕竟一个人的力量总是有限的。

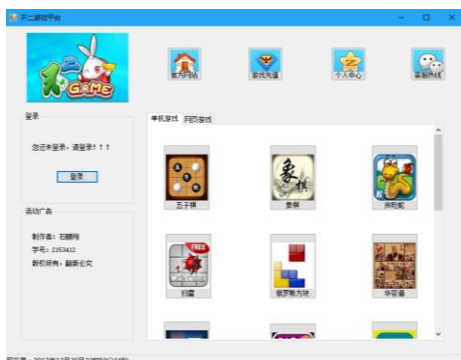


图 24 未优化的图



图 25 优化后的图