

# 课 程 设 计 报 告

课程名称 C++面向对象程序设计

课题名称 公司人事管理系统、贪吃蛇

班级 无 63

学号 2016011058

姓名 漆耘含

2017 年 7 月 7 日

## 目录

1. 公司人事管理系统.....	3
1.1 系统需求分析.....	3
1.2 总体设计.....	4
1.3 详细设计.....	5
1.4 系统调试.....	11
1.5 结果分析.....	11
2. 贪吃蛇大作战.....	14
2.1 系统需求分析.....	14
2.2 总体设计.....	14
2.3 详细设计.....	16
2.4 系统调试.....	21
2.5 结果分析.....	22
3. 总结.....	26
附录：源程序清单.....	27
公司人事管理系统源程序.....	27
贪吃蛇大作战源程序.....	56
进度安排.....	83
评分表.....	84

# 1. 公司人事管理系统

## 1.1 系统功能需求分析

公司设有很多个岗位，有 CEO，经理，销售，技工等等，人事管理需要有员工的个人信息，比如姓名、性别，当员工进入公司之后，也会分配工号（即编号）和相应的等级。当一个人的信息发生变化的时候，比如升职，那么需要对员工的信息进行修改，当一年之后，需要对整个公司的基础薪水改变，那么就需要人为进行改变并且同步到数据库中。因此该人事管理系统有以下 5 个功能：

数据录入功能。通过人为地输入，将信息全部保存在数据库中。在进行数据录入的时候，会有相应的选择，比如是选择录入 CEO 还是经理，并且在每一次录入完成之后，都会询问是否继续录入信息，如果没有，则返回菜单。

数据总览功能。这个功能是能够显示全部的人员数据。方便管理层进行审阅。在显示全部信息之后，返回菜单。

个人信息查询功能。当不需要进行员工总览的时候，就可以通过该功能进行查询。查询方式分为两个，分别是通过名字和工号来查询。在每一次查询之后，会询问是否继续进行查询，如果否，则返回目录。

个人信息修改功能。这个功能是通过编号来进行修改的，首先输入该员工的工号，然后进行重新输入，修改之后会给出修改成功的提示，并且将修改后的信息保存到原来的数据库中。

薪水修改功能。当需要对员工的基础薪水进行调整的时候，先选择要调整哪一个职位的薪水，并且在调整之后，会有询问是否马上更新到数据库中。在一个循环结束之后，有是否继续修改的询问，如果否，则返回目录。

整个系统以目录的形式来运行，用户可以自由选择不同的功能。并且在执行完一个功能之后，会回到目录，直到退出系统。

1.2 总体设计

人事管理系统总共包括五个功能：数据录入、数据总览、个人信息查询、个人信息修改、基础薪水修改。不同职工的信息是不一样的，都具备的信息是姓名、性别、等级、编号、薪水。其中薪水的计算方式不一样。CEO 和经理具有的是固定工资，技工具有的是每月工作小时数，销售是每月销售总额和提成比例，销售经理有固定工资和销售提成。

在数据录入时，会有相应的选择，比如是选择录入 CEO 还是经理，在每一次录入完成之后，都会询问是否继续录入信息，如果没有，则返回菜单。并且每一个职位的输入格式都是不一样的，需要根据提示来进行输入。

在数据总览时，会直接显示所有员工的信息（按照工号进行升序排列）

在个人信息查询时，查询方式分为两个，分别是通过名字和工号来查询。在每一次查询之后，会询问是否继续进行查询，如果否，则返回目录。

在个人信息修改时，首先输入该员工的工号，然后进行重新输入（按照提示来输入），修改之后会给出修改成功的提示，并且将修改后的信息保存到原来的数据库中。

在基础薪水修改时，先选择要调整哪一个职位的薪水，并且在调整之后，会询问是否马上更新到数据库中。在一个循环结束之后，有是否继续修改的询问，如果否，则返回目录。

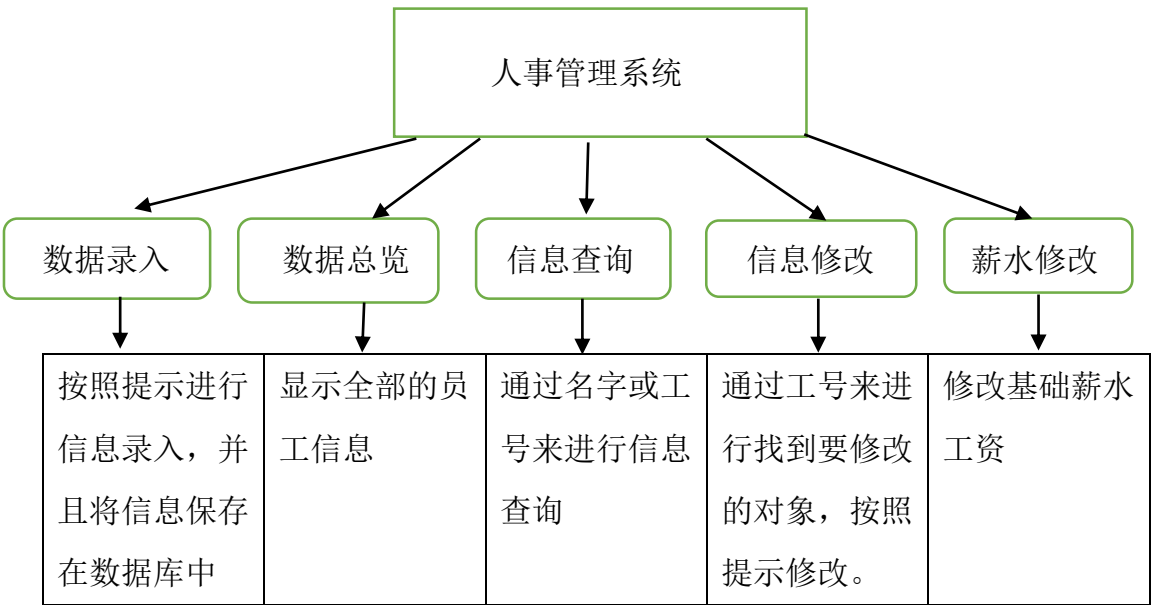


图 1 人事管理系统功能总览

### 1.3 详细设计

人事管理系统工有 6 个类：

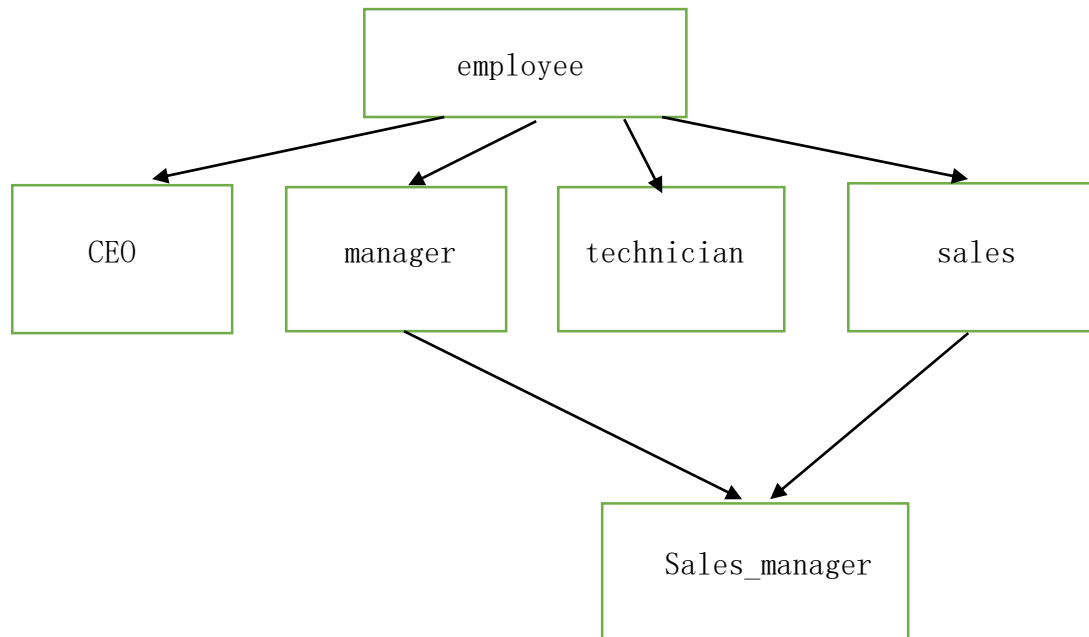


图 2 类层次视图

人事管理系统功能模块：

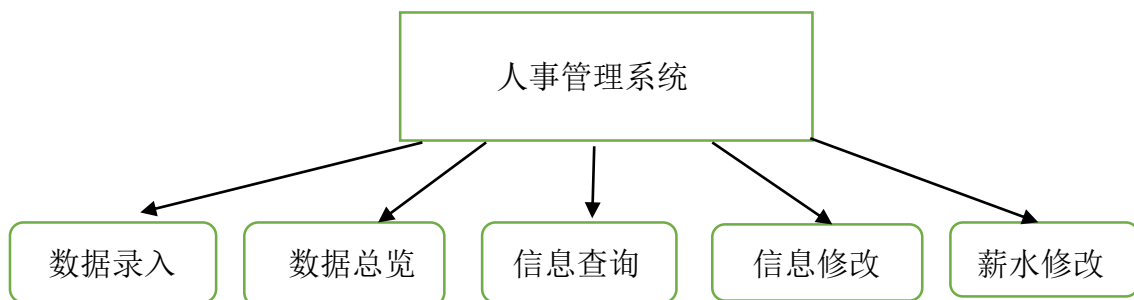


图 3 系统功能视图

a) 数据录入

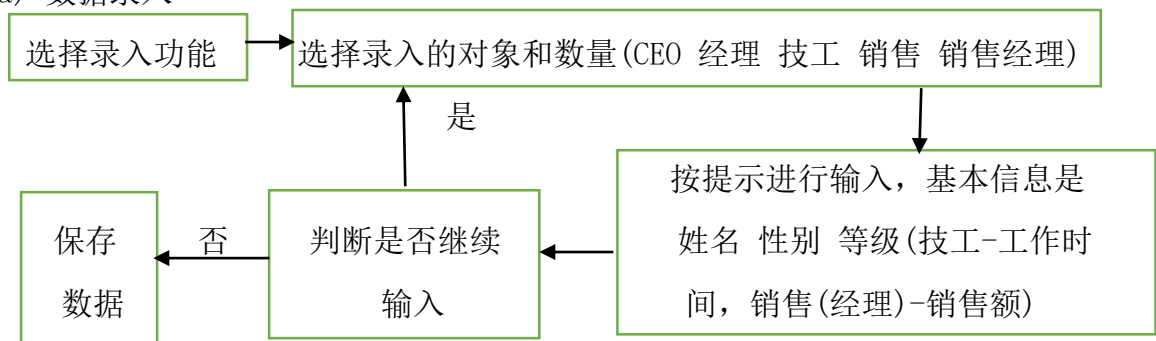


图4 数据录入模块

b) 数据总览

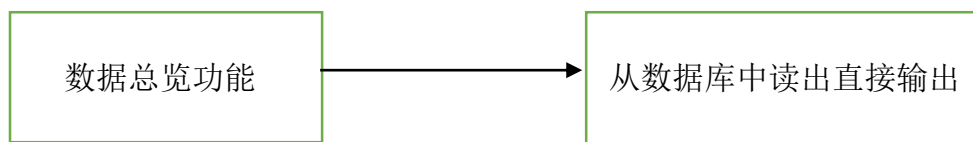


图5 数据总览模块

c) 信息查询

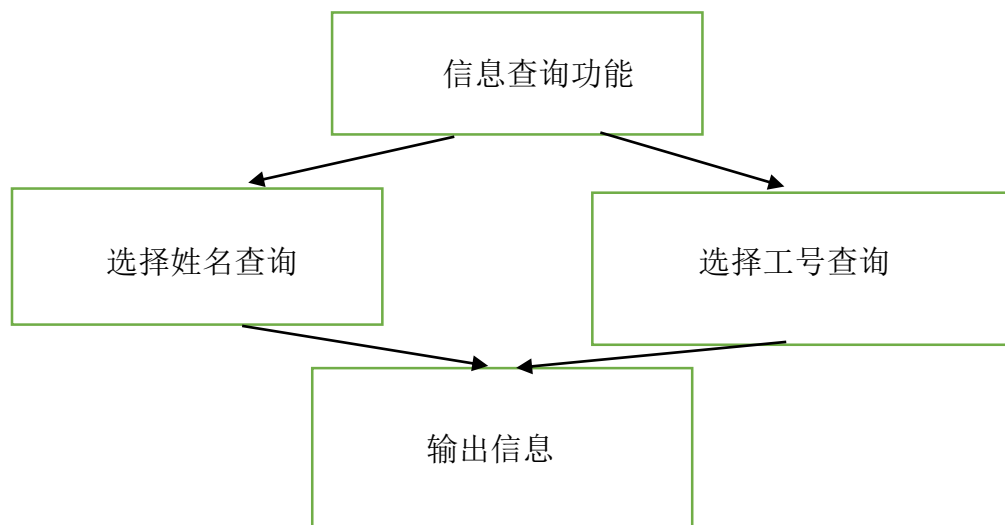


图6 信息查询模块

d) 信息修改

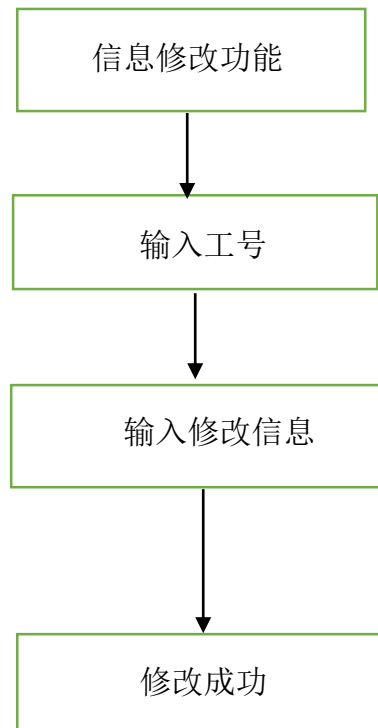


图 7 信息修改模块

e) 修改薪资

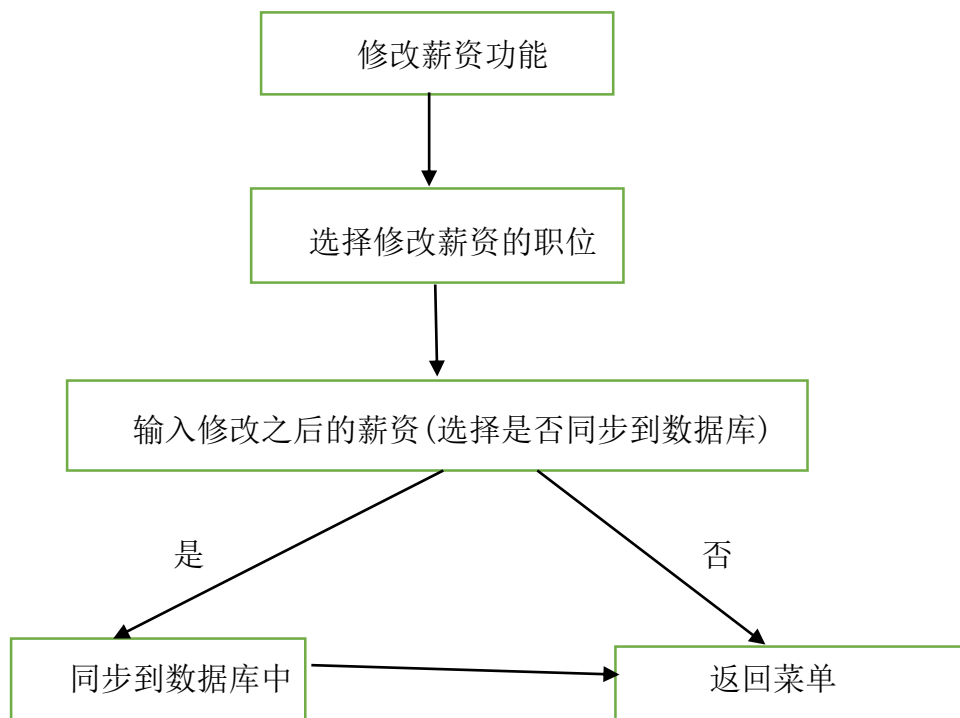
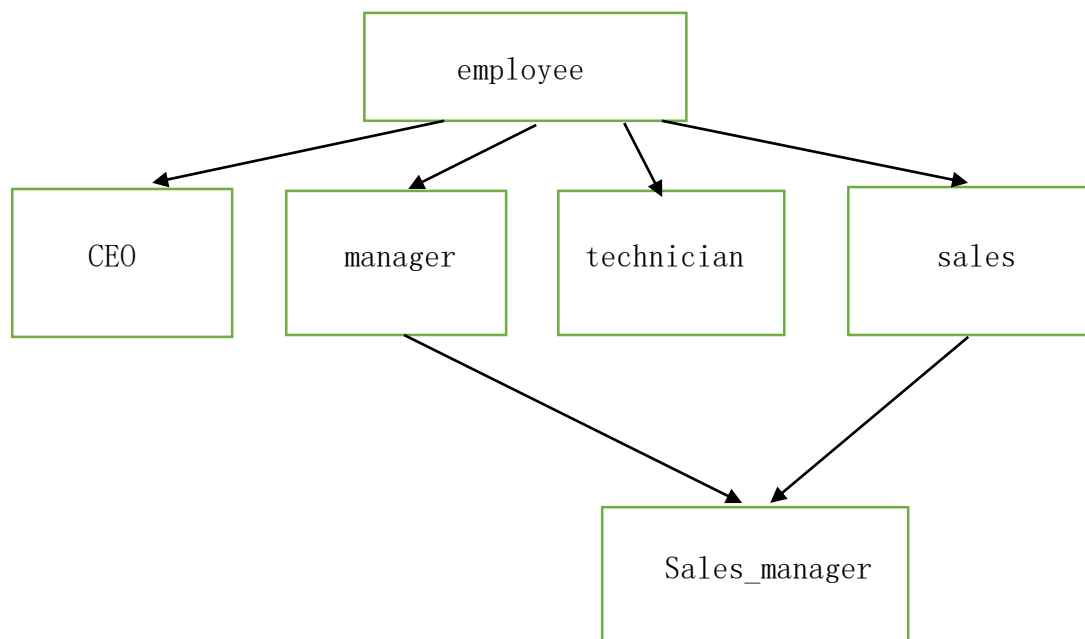


图 8 修改薪资函数

公司人事管理系统 6 个类的 UML 图：



具体类实现：

employee 类：

```
public:
    virtual double pay_calculate()=0; (计算月薪函数)
    virtual void adjust_accumay(){}; (调整月薪函数)
    static int num;
protected:
    char name[10]; (姓名)
    char sex[3]; (性别)
    int NO; (编号)
    char grade; (级别)
    double accumpay; (月薪)
```



CEO 类:

```
Public:

    virtual double pay_calculate();

    friend ostream & operator<<(ostream & output, CEO & C);

    friend istream & operator>>(istream & input, CEO & C);

    virtual void adjust_accumpay();

    double static get_accumpay_c(); (得到 CEO 的月薪函数)

Private:

    static double accumpay_CEO; (CEO 的月薪)
```

manager 类:

```
Public:

    virtual double pay_calculate();

    friend ostream & operator<<(ostream & output, manager& m);

    friend istream & operator>>(istream & input, manager & m);

    virtual void adjust_accumpay();

    double static get_accumpay_m(); (得到经理现在月薪)

    double static get_accumpay_mpre(); (得到经理以前的月薪)

Protected:

    static double accumpay_manager; (经理现在的月薪)

    static double accumpay_manager_pre; (经理以前的月薪)
```

technician 类:

```
Public:

    virtual double pay_calculate();

    friend ostream & operator<<(ostream & output, technician & t);

    friend istream & operator>>(istream & input, technician & t);

    virtual void adjust_accumpay();

    double static get_accumpay_t(); (得到现在技工每小时工资)
```

```
double static get_accumpay_tpre(); (得到以前每小时工资)
Private:
    int work_hour; (工作的时间)
    static double accumpay_per_hour; (现在每小时工资)
    static double accumpay_per_hour_pre; (以前每小时工资)
```

sales 类:

```
Public:
    virtual double pay_calculate();
    friend ostream & operator<<(ostream & output, sales & s);
    friend istream & operator>>(istream & input, sales & s);
    virtual void adjust_accumpay();
    double static get_accumpay_S();
    double static get_accumpay_Spre();
Protected:
    double amount; (销量)
    double static propotion_sales;
    double static propotion_sales_pre;
```

sales\_manager 类:

```
Public:
    virtual double pay_calculate();
    friend ostream & operator<<(ostream & output, sales_manager &s);
    friend istream & operator>>(istream & input, sales_manager &s);
    virtual void adjust_accumpay();
    double static get_accumpay_SM();
    double static get_accumpay_SMpre();
Private:
    double static propotion_sales_manager;
    double static propotion_sales_manager_pre;
```

# 1.4 系统调试

在程序编写完成之后，我进行了程序调试。

在进行数据录入之后，进行数据信息全部输出，但发现输出全是“烫烫……”，这也就是说要么数据根本没有保存到文件里面，要么就是从文件中读的时候没有成功。我打开了目标文件中的“数据库.txt”，发现里面已经有相应的数据，这就说明是在读出的时候没有成功。回去检查代码的时候，发现在使用 getline 函数的时候，并没有设置读的位数，并且判断截止符号并没有与数据库中的间隔相对应，这就导致了没有能够成功从文件中读出。

在进行信息查询的时候，通过工号查询没有成功，回去看代码的时候，发现从文件中读出来的编号是字符串，而输入的是一个整型数，根本不能匹配，于是我使用了函数库中的 atoi 函数，让他们的类型相同，从而能够进行匹配。

在进行修改函数的时候，发现每一次修改完之后，都不能回到菜单，后来发现判断是否继续输入的是 Y/N，而我为了防止错误输入，加了一个 while 循环，如果输入的不是 Y/N，那么就一直输入，我的 while 函数的判断条件却写的是 A/B，这就是问题的所在。

在进行薪资修改的时候，发现同步数据的时候，不能同步正常的数据进去，这是因为在从文件中读出的信息只有最后的工资，没有他们的工时薪资等信息，因此我就在每一个类中添上了一个成员，用于保存每一次修改之前的薪资，并添上相应的得到薪资函数，这样在进行薪资更新的时候，就能够计算出正确的更新过后的薪资了。

# 1.5 测试结果与分析

测试文件：

CEO	漆松含	男	1	A	15000	
经理	徐远帆	男	2	B	15000	
技工	刘树清	男	3	C	12000	
技工	詹明韬	男	4	C	18000	
销售	谭博文	男	5	D	10000	
销售经理	刘杰	男	6	B	15200	

测试结果：

```
-----
:                                     :
:      耘含企业人事管理系统      :
:      欢迎使用                    :
:                                     :
-----
1. 员工数据录入
2. 数据库总览
3. 员工信息查询
4. 员工数据修改
5. 薪水调整
6. 退出
*****
您想要访问的序号:
2
```

```
-----
:                                     :
:      耘含有限公司员工总览      :
:                                     :
-----
职位:CEO
姓名:漆耘含
性别:男
编号:1
等级:A
月薪:20000

职位:经理
姓名:徐远帆
性别:男
编号:2
等级:B
月薪:15000

职位:技工
姓名:刘树清
性别:男
编号:3
等级:C
月薪:12000

职位:技工
姓名:詹明韬
性别:男
编号:4
等级:C
月薪:18000

职位:销售
姓名:谭博文
性别:男
编号:5
等级:D
月薪:10000

职位:销售经理
姓名:刘杰
```



## 2. 贪吃蛇大作战

### 2.1 系统需求分析

贪吃蛇主要是实现游戏功能,游戏模块分为两个功能,一个是匀速游戏模块,一个是加速游戏模块。其中匀速游戏模块共有四个等级可供选择。除了游戏功能之外,还有创建游戏账号、阅读游戏规则、英雄榜排名的功能。

创建游戏账号是玩家输入游戏昵称,并保存在数据库中。

阅读游戏规则,选择该功能后,仔细阅读相应的游戏规则,并且严格按照游戏规则来进行游戏。

匀速贪吃蛇有四个速度可供选择,在一次游戏结束之后,会出现所得分数和所用时间。并且会把成绩保存在数据库中。

变速贪吃蛇是根据所得分数来进行速度的提升,当所得分数越高,速度会越快,当速度达到某一个水平之后,会保持匀速进行。在游戏结束之后,同样也会出现所得分数和所用时间。并且把成绩保存在数据库中。

英雄榜功能是排列数前三名的成绩和相应的玩家,并且会显示所有玩家一共得到的分数。

### 2.2 总体设计

贪吃蛇这个游戏一共包含有五个大的功能,分别是创建游戏账户,阅读游戏规则,匀速贪吃蛇游戏,变速贪吃蛇游戏,英雄榜。

创建游戏账户,是用户在窗口上自己用键盘输入昵称,并且按回车进行确认。在创建用户之后,系统会把账户和分数写进数据库中,其中分数为0。

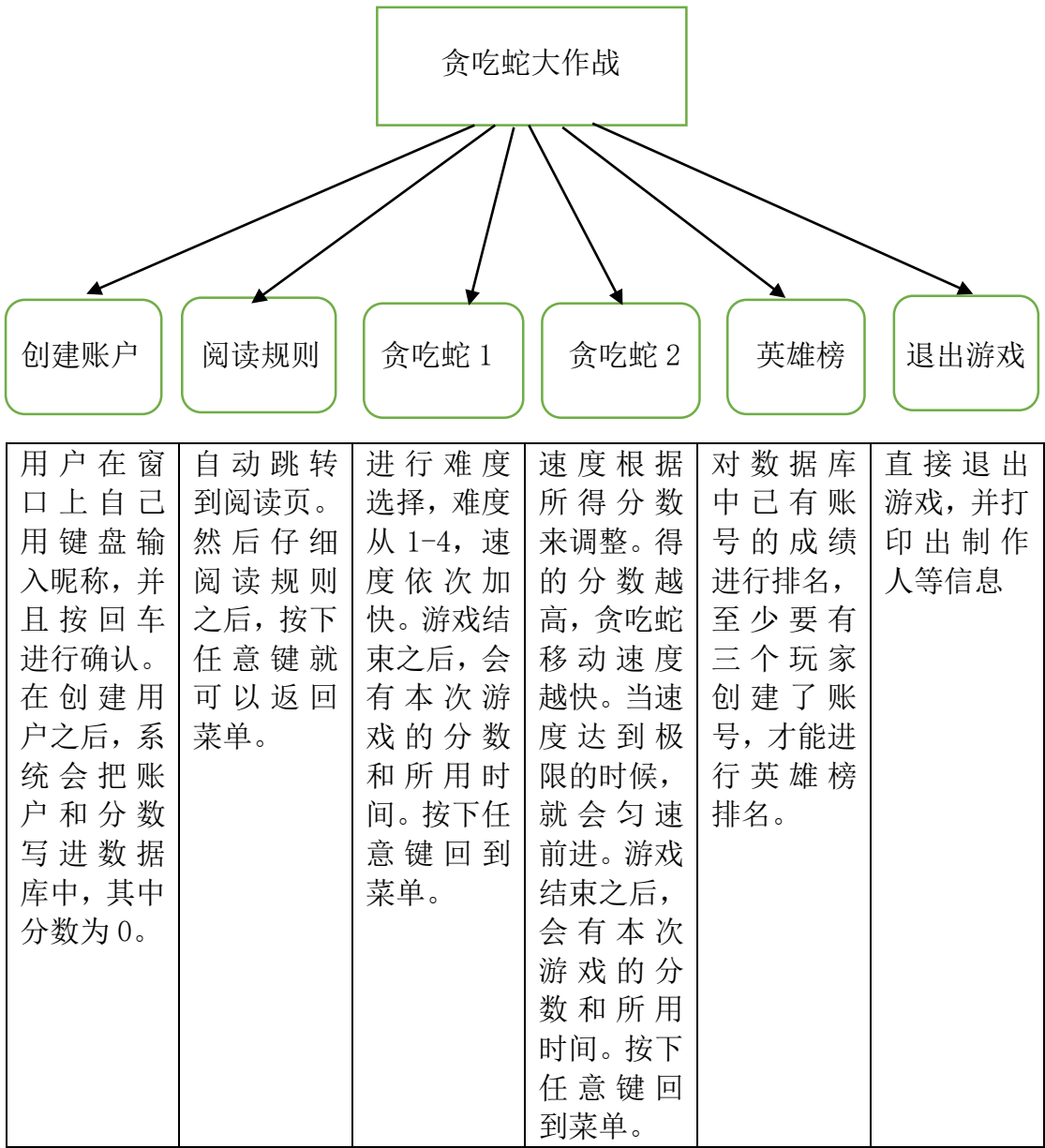
阅读游戏规则模块,在菜单中选定该功能之后,会自动跳转到阅读页。然后仔细阅读规则之后,按下任意键就可以返回菜单。

在进行匀速贪吃蛇游戏的时候,首先会跳转到一个页面,进行难度选择,难度从1-4,速度依次加快。在贪吃蛇游戏中,要让贪吃蛇动起来,需要在界面上一遍一遍地刷,当一个画面结束之后,立马清空界面,马上画出下一张图,这样人眼看到的图画就是动态的。在判定贪吃蛇游戏结束之后,会出现一个画面,上面会打印出“DEAD”的字样,并且下面会有本次游戏的分数和所用时间。按下任意键回到菜单。并且会把相应的分数保存在数据库对应的账号的成绩里面

在进行变速贪吃蛇游戏的时候,直接就开始游戏,速度根据所得分数来调整。得的分数越高,贪吃蛇移动速度越快。当速度达到极限的时候,就会匀速前进。同时,在判定贪吃蛇游戏结束之后,也同样会出现一个画面,上面会打印出“DEAD”的字样,并且下面会有本次游戏的分数和所用时间。按下任意键回到菜单。并且会把相应的分数保存在数据库对应的账号的成绩里面

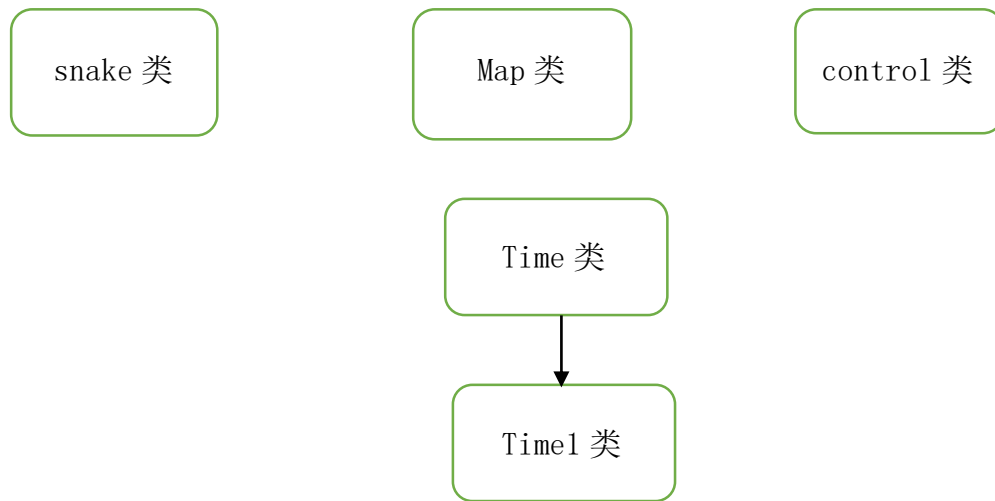
英雄榜是能够对数据库中已有账号的成绩进行排名。把数据从数据库中读出保存下来,然后进行冒泡排序,最后输出前三名。这也就意味着,至少要有三个玩家创建了账号,才能进行英雄榜排名。

贪吃蛇功能模块图:

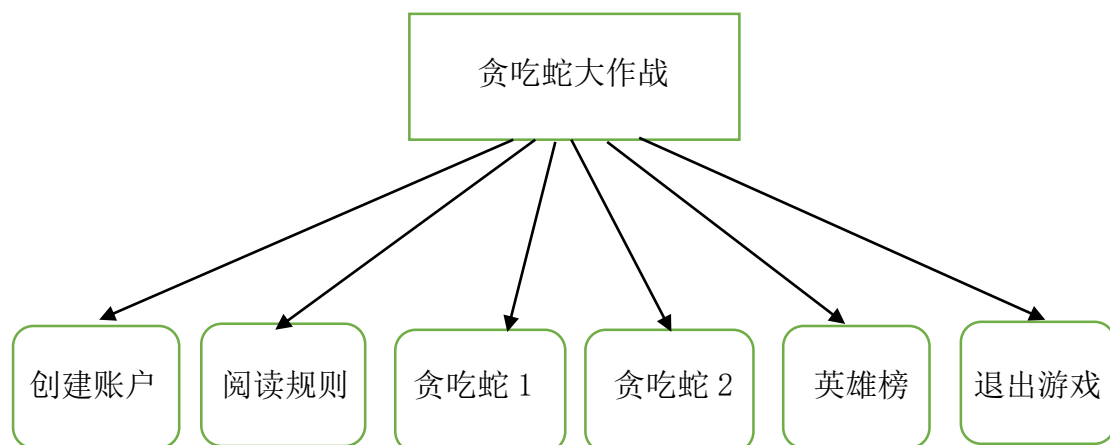


## 2.3 详细设计

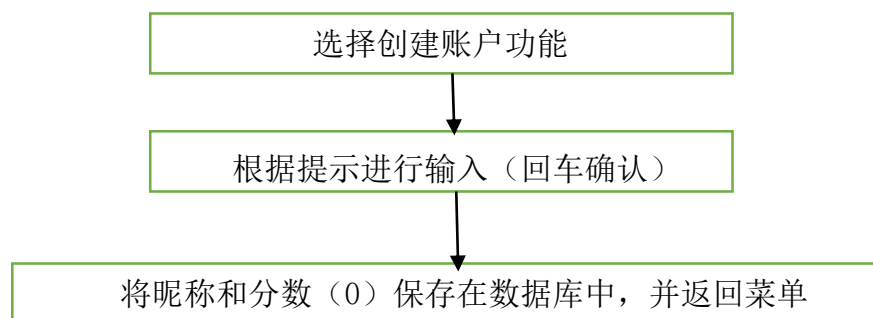
贪吃蛇的类层次图：



贪吃蛇各个功能模块的实现：

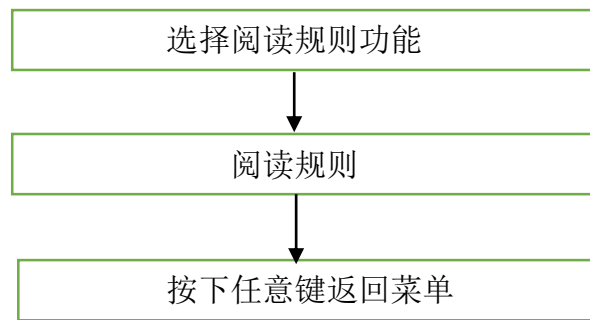


### 3.1 创建账户功能

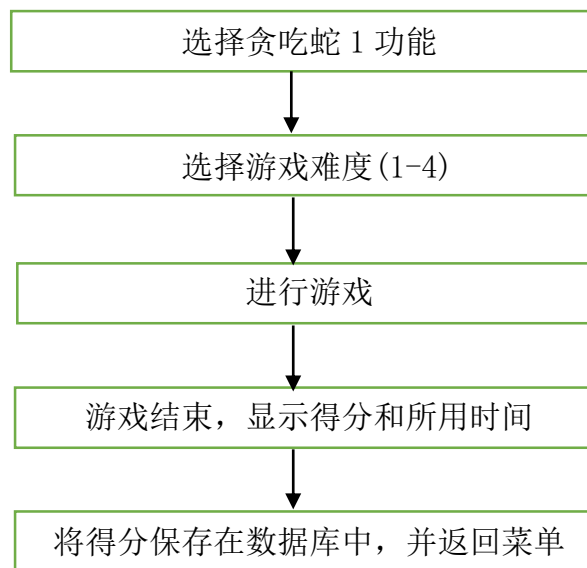




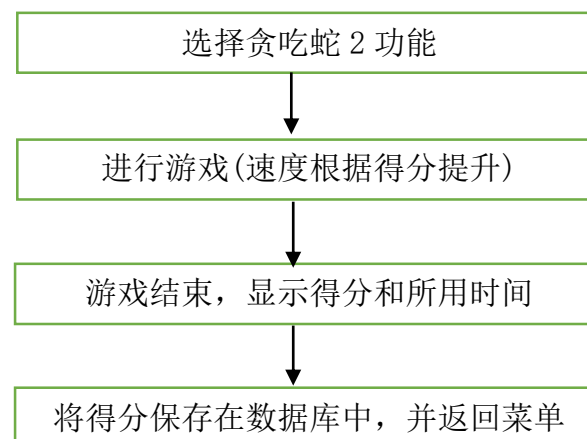
### 3.2 阅读规则功能



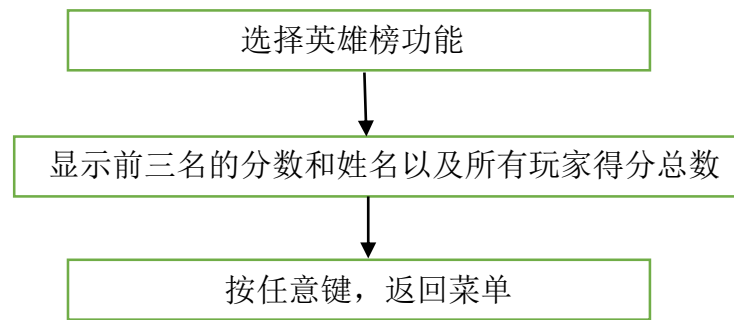
### 3.3 贪吃蛇 1 功能模块



### 3.4 贪吃蛇 2 功能模块



### 3.5 英雄榜

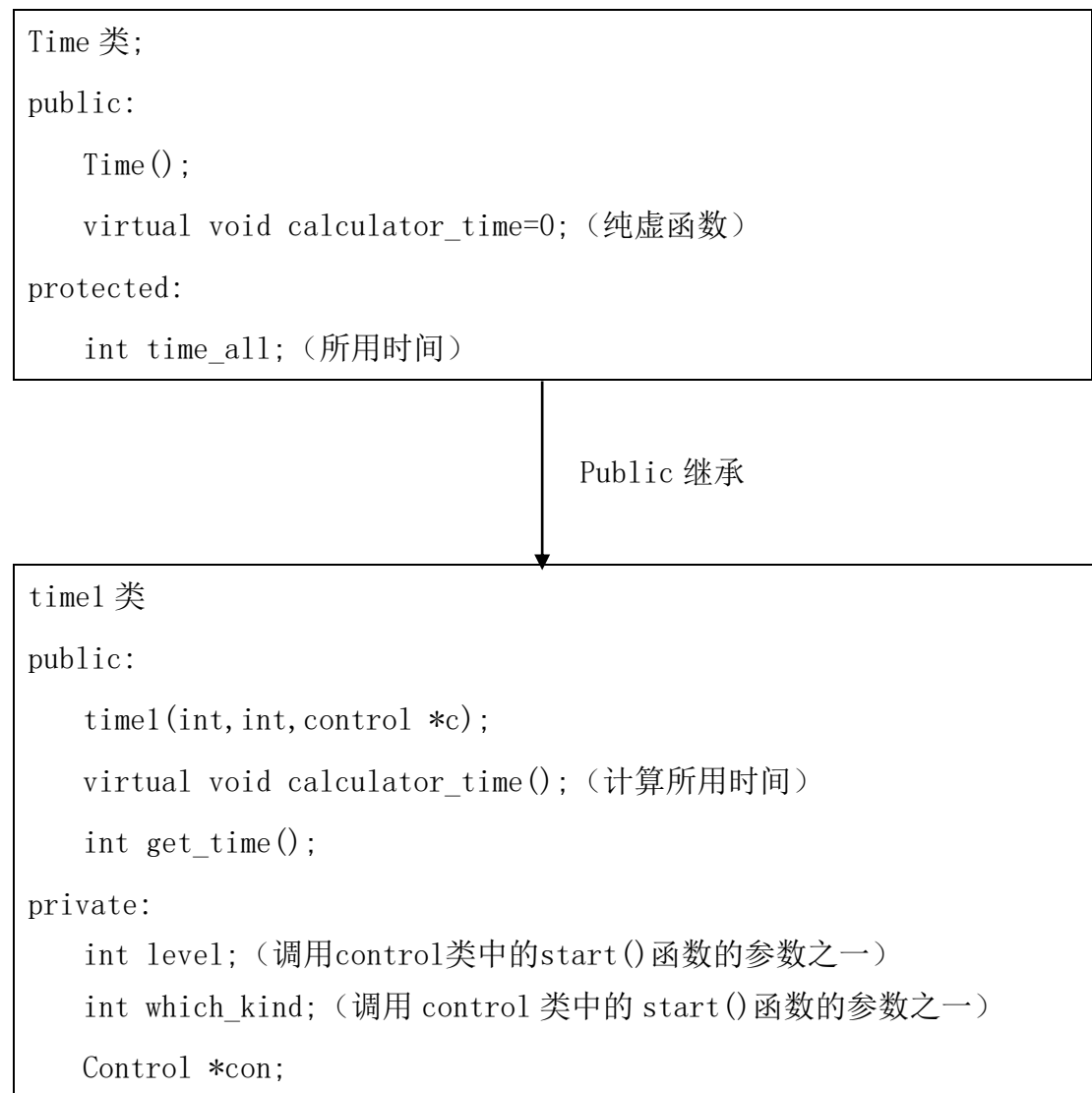


五个类的具体实现：

```
Map 类：
public:
    Map(int, int); (构造函数)
    void draw(); (画地图)
    void set_food(); (设置食物坐标)
    void draw_food(); (画出食物)
    int get_food_x(); (得到食物的 x 坐标)
    int get_food_y(); (得到食物的 y 坐标)
    int get_Map_length(); (得到地图的长度)
    int get_Map_width(); (得到地图的宽度)
private:
    int Map_length; (地图长度)
    int Map_width; (地图宽度)
    int food_x; (食物 x 坐标)
    int food_y; (食物 y 坐标)
```

```
snake 类:
public:
    snake(const tint, int, Map *map); (构造函数)
    void draw(); (画出蛇身)
    void move_left(); (向左移动)
    void move_right(); (向右移动)
    void move_up(); (向上移动)
    void move_down(); (向下移动)
    void move_forward(); (向上移动)
    int eat_food(); (吃到食物)
    bool judge_life(); (判断蛇的生命)
    bool get_life(); (得到蛇的生命状态)
    char get_head_direction(); (得到当前前进方向)
    void change_head_direction(char); (改变蛇的前进方向)
    Map *m;
    void change_life(bool); (改变蛇的生命状态)
private:
    int body_x[100]; (蛇身 x 坐标)
    int body_y[100]; (蛇身 y 坐标)
    int snake_length; (蛇的长度)
    bool life; (生命状态)
    char head_direction; (前进方向)
```

```
control 类:
public:
    control(snake *s);
    void start(int, int); (游戏开始)
    void game_over(); (游戏结束)
    bool left(); (向左移动)
    bool right(); (向右移动)
    bool up(); 向上移动
    bool down(); (向下移动)
    void show_score(); (所得分数)
    void forward(); (向上移动)
    int operator +(int); (对+进行重载)
private:
    char direction;
    Direction D; (前进方向)
    snake *sna;
    int score; (所得分数)
```



## 2.4 系统调试

在整个贪吃蛇游戏的调试中，出现最多问题的就是在两个游戏模块部分。

首先，在进行 MAP 类画地图和画食物的过程中，会在一张图上出现很多个食物，后来发现这是因为在重新绘制食物的时候，没有刷新，也就是没有先刷新界面，然后再画食物，所以如果没有刷新界面的话，那么一张地图上就会显示出多个食物。

其次，就是在对贪吃蛇移动时候出现了很多 bug。对于贪吃蛇移动的想法，就是后一节的坐标等于前一节的坐标，然后蛇头的坐标根据发出的指令来进行调整。然后清空界面，重新绘制地图食物和蛇身。再具体实现的时候，发现在进行

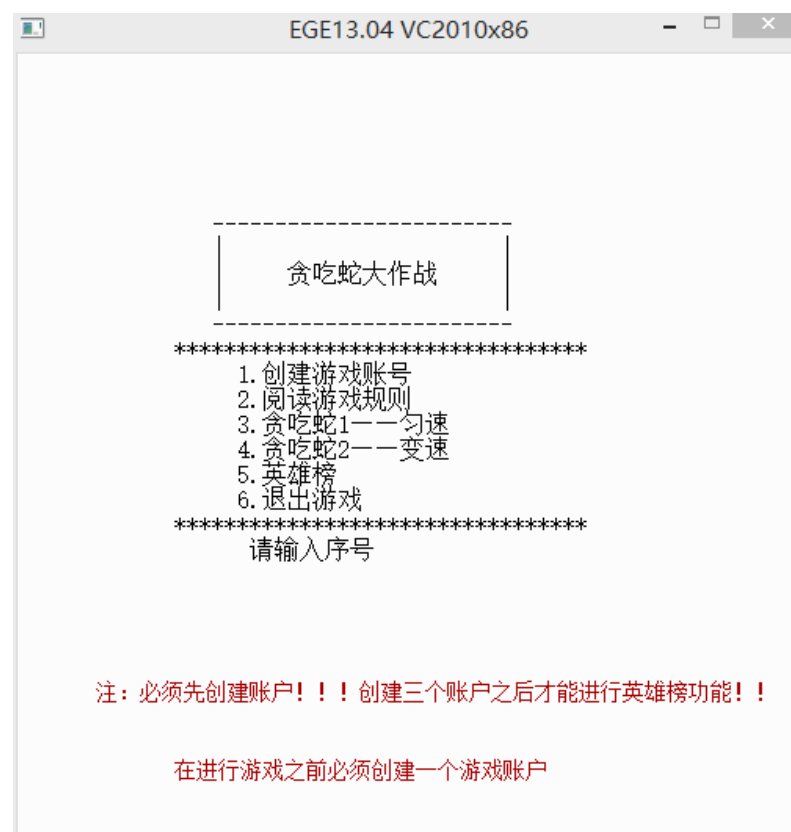
操作的时候，蛇变成了一个实心圆，后来发现，这是因为在对蛇坐标进行变换的时候，先从蛇头开始变化了，所以导致后面蛇身的坐标全部都等于蛇头的坐标，从而在地图上显示出来就是一个点。在实现的时候应该从蛇尾先开始，最后再对蛇头坐标进行变换。

然后是在判断蛇是否死亡的时候，一共有两种情况，即一种是碰到了边界。另一种是碰到了蛇本身（包括对蛇下向与当前前进方向相反的方向前进的指令）。因为涉及的蛇是半径为 5 的一个圆，那么在进行判断的时候，需要判断的是两个点之间的距离是否小于 5，而我在第一次实现的时候，判断条件是判断两个点的坐标是否相同，那么这就无法进行判断是否撞墙或者碰到自己的身体。

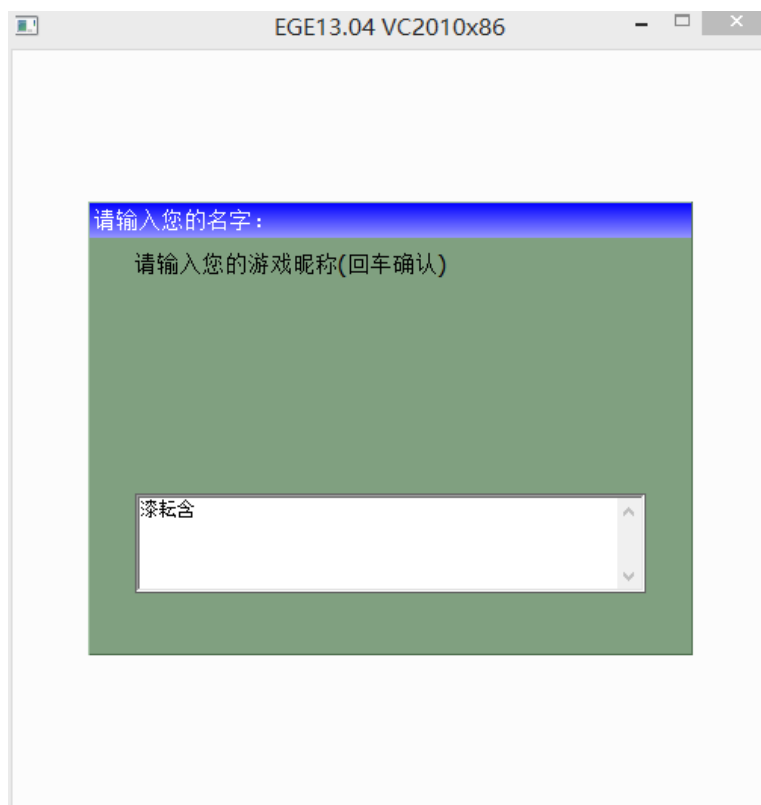
最后是在进行英雄榜功能的时候，发现在进行完一次游戏之后，发现原来的数据全部都消失不见了，之后最新的一次数据。经过检查发现是因为文件的打开方式不对，打开文件的时候把原来的数据全部删除了。

## 2.5 结果分析

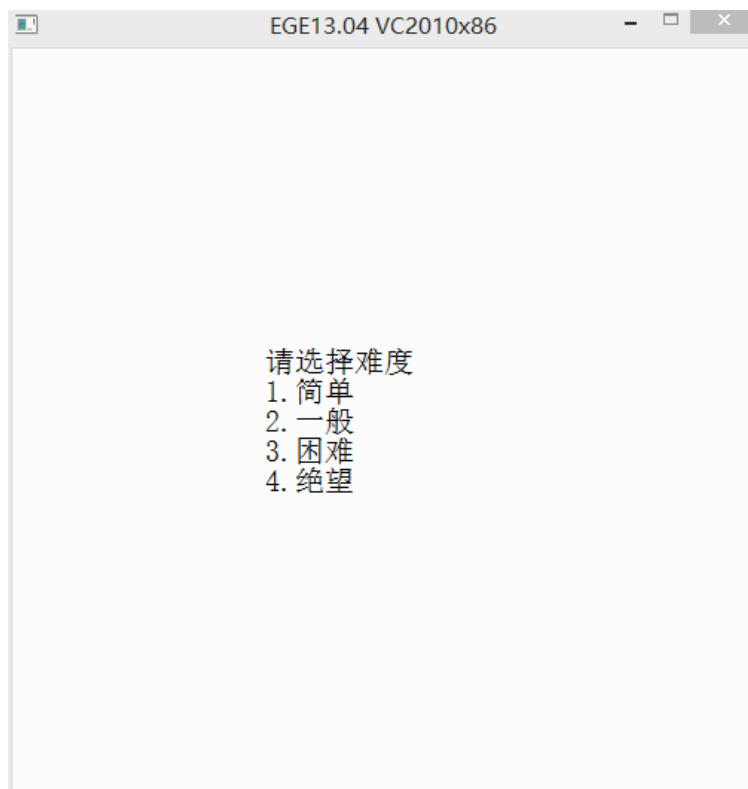
菜单功能：



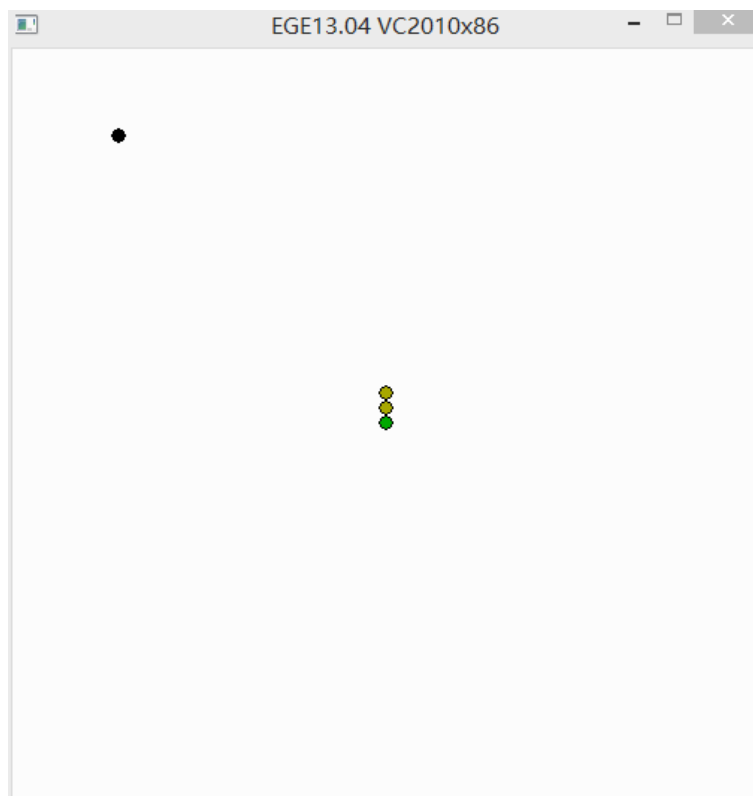
功能 1 界面：



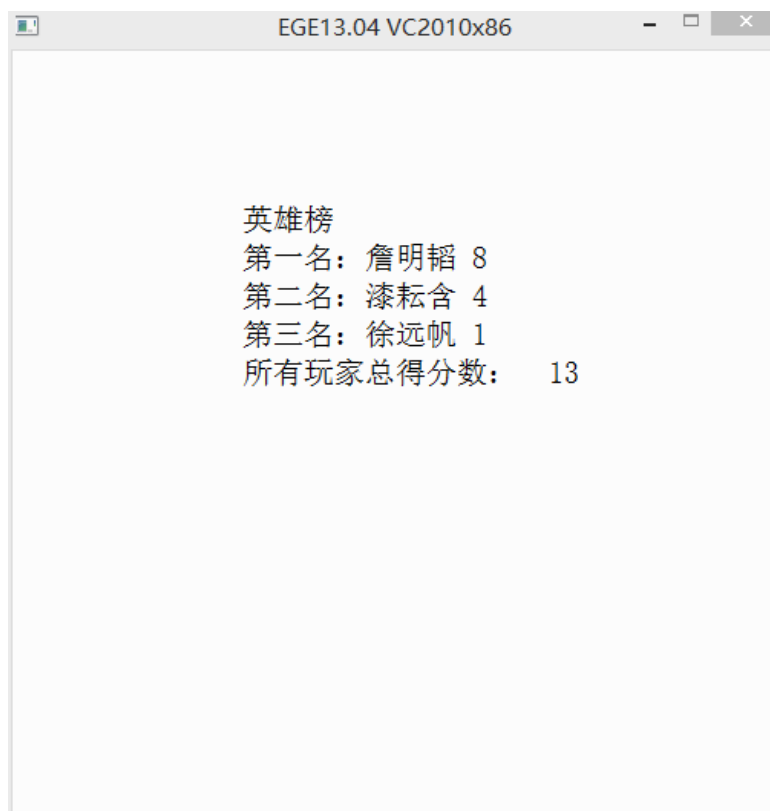
贪吃蛇 1 选择界面：



贪吃蛇游戏界面：



英雄榜界面：





结束界面：



贪吃蛇游戏有一个小问题，就是贪吃蛇和食物无法完全对齐，判断是否吃到食物的条件是蛇头的圆心和食物的圆心之间距离小于 5，这个是可以理解的，但是在图形界面上表现出来，就有一点奇怪的感觉，因为有一些错位，这是可以继续提升的地方。

### 3. 总结

在编写“人事管理系统”过程中，碰到了很多问题，自己通过看书或者跟同学交流，能够很好的解决这些问题，自己的能力得到了提高。总的来说，这个系统中中规中矩，没有什么特点，都是很普通的功能，在编写代码的时候，编写的习惯一定要培养好，不然自己写出来自己都看不懂。另外在这个程序中，有一个程序段被频繁使用，但是没法编写成一个函数（功能不允许），因此重复了好几次，我觉得可以从这上面继续提高。

在编写贪吃蛇的时候，刚开始没有头绪，不知道怎么去设计，后来搜集了很多资料，慢慢地摸清楚了套路，后来自己也设计出来了贪吃蛇。在编写的时候，使用了 EGE 编程，这是跟普通的控制台程序不一样了，这是这个贪吃蛇比较好的一点，它另外开了一个界面。但这用也会有问题，比如无法很好地对齐等问题。

和上学期相比，这学期能够很好的运用知识来编写一个较大的程序，这是很大的进步，但同时，我又认识到自己能力的不足，还有很长的路要走。

## 附录：源程序清单

### 公司人事管理系统：

“employee.h”

```
class employee
{
    public:
        employee() {} ;
        virtual double pay_calculate()=0;
        virtual void  adjust_accumpay() {} ;
        static int num;
    protected:
        char name[10];
        char sex[3];
        int NO;
        char grade;
        double accumpay;
};

int employee::num=1;

class CEO:public employee
{
    public:
        CEO();
        CEO(char n[], char s[], char g);
        virtual double pay_calculate();
        friend ostream & operator<<(ostream & output,CEO & C);
        friend istream & operator>>(istream & input,CEO & C);
        virtual void adjust_accumpay();
        double static get_accumpay_c();
```

```

    private:
        static double accumpay_CEO;
};
double CEO::accumpay_CEO=20000;
CEO::CEO()
{
    NO=num;
    num=num+1;
}
CEO::CEO(char n[], char s[], char g)
{
    strcpy(name, n);
    strcpy(sex, s);
    grade=g;
    NO=num;
    num=num+1;
}
double CEO::pay_calculate()
{
    accumpay=accumpay_CEO;
    return accumpay;
}
ostream & operator<<(ostream & output, CEO & C)
{
    output<<"CEO"<<' \t' <<C.name<<' \t' <<C.sex<<' \t' <<C.NO<<' \t' <<C.grade<<' \t' <<C.accumpay_CEO<<' \n' ;
    return output;
}
istream & operator>>(istream & input, CEO & C)

```

```

{
    cout<<"请依次输入姓名、性别、等级:"<<' \n' ;
    input>>C.name>>C.sex>>C.grade;
    return input;
}

void CEO::adjust_accumpay()
{
    cout<<"输入 CEO 工资:";
    double a;
    cin>>a;
    accumpay_CEO=a;
}

double CEO::get_accumpay_c()
{
    return accumpay_CEO;
}

class manager:virtual public employee
{
    public:
        manager();
        manager(char n[],char s[],char g);
        virtual double pay_calculate();
        friend ostream & operator<<(ostream & output,manager & m);
        friend istream & operator>>(istream & input,manager & m);
        virtual void adjust_accumpay();
        double static get_accumpay_m();
        double static get_accumpay_mpre();
    protected:
        static double accumpay_manager;

```

```

        static double accumpay_manager_pre;
};
double manager::accumpay_manager=15000;
double manager::accumpay_manager_pre=15000;
manager::manager()
{
    NO=num;
    num=num+1;
}
manager::manager(char n[], char s[], char g)
{
    strcpy(name, n);
    strcpy(sex, s);
    grade=g;
    NO=num;
    num=num+1;
}
double manager::pay_calculate()
{
    accumpay=accumpay_manager;
    return accumpay;
}
ostream & operator<<(ostream & output, manager & m)
{
    output<<"经理
"<<' \t' <<m. name<<' \t' <<m. sex<<' \t' <<m. NO<<' \t' <<m. grade<<' \t' <<m. accu
mpay_manager<<' \n' ;
    return output;
}

```

```

istream & operator>>(istream & input, manager & m)
{
    cout<<"请依次输入姓名、性别、等级:"<<' \n';
    input>>m.name>>m.sex>>m.grade;
    return input;
}

void manager::adjust_accumpay()
{
    cout<<"输入经理工资:";
    double a;
    cin>>a;
    accumpay_manager_pre=accumpay_manager;
    accumpay_manager=a;
}

double manager::get_accumpay_m()
{
    return accumpay_manager;
}

double manager::get_accumpay_mpre()
{
    return accumpay_manager_pre;
}

class technician:virtual public employee
{
public:
    technician();
    technician(char n[], char s[], char g, int w);
    virtual double pay_calculate();
    friend ostream & operator<<(ostream & output, technician & t);
}

```

```

        friend istream & operator>>(istream & input, technician & t);
        virtual void adjust_accumpay();
        double static get_accumpay_t();
        double static get_accumpay_tpre();
    private:
        int work_hour;
        static double accumpay_per_hour;
        static double accumpay_per_hour_pre;
};

double technician::accumpay_per_hour=30;
double technician::accumpay_per_hour_pre=30;
technician::technician()
{
    NO=num;
    num=num+1;
}

technician::technician(char n[], char s[], char g, int w)
{
    strcpy(name, n);
    strcpy(sex, s);
    grade=g;
    NO=num;
    num=num+1;
    work_hour=w;
    accumpay=pay_calculate();
}

double technician::pay_calculate()
{
    double a;

```



```

        a=accumpay_per_hour*work_hour;

        return a;
    }

ostream & operator<<(ostream & output, technician & t)
{
    output<<"技工

    "<<' \t' <<t.name<<' \t' <<t.sex<<' \t' <<t.NO<<' \t' <<t.grade<<' \t' <<t.accu
    mpay<<' \n' ;

    return output;
}

istream & operator>>(istream & input, technician & t)
{
    cout<<"请依次输入姓名、性别、等级、工作时间:"<<' \n' ;
    input>>t.name>>t.sex>>t.grade>>t.work_hour;

    t.accumpay=t.pay_calculate();

    return input;
}

void technician::adjust_accumpay()
{
    cout<<"请输入技工每小时的工资:";

    double a;

    cin>>a;

    accumpay_per_hour_pre=accumpay_per_hour;

    accumpay_per_hour=a;
}

double technician::get_accumpay_t()
{
    return accumpay_per_hour;
}

```

```

double technician::get_accumpay_tpre()
{
    return accumpay_per_hour_pre;
}

class sales:virtual public employee
{
    public:
        sales();
        sales(char n[], char s[], char g, double a);
        virtual double pay_calculate();
        friend ostream & operator<<(ostream & output, sales & s);
        friend istream & operator>>(istream & input, sales & s);
        virtual void adjust_accumpay();
        double static get_propotion_S();
        double static get_propotion_Spre();
    protected:
        double amount;
        double static propotion_sales;
        double static propotion_sales_pre;
};

double sales::propotion_sales=0.05;
double sales::propotion_sales_pre=0.05;
sales::sales()
{
    NO=num;
    num=num+1;
}

sales::sales(char n[], char s[], char g, double a)
{

```

```

        strcpy(name, n);
        strcpy(sex, s);
        grade=g;
        NO=num;
        num=num+1;
        amount=a;
        accumpay=pay_calculate();
    }

double sales::pay_calculate()
{
    double a;
    a=amount*propotion_sales;
    return a;
}

ostream & operator<<(ostream & output, sales & s)
{
    output<<"销售
    "<<' \t' <<s.name<<' \t' <<s.sex<<' \t' <<s.NO<<' \t' <<s.grade<<' \t' <<s.accu
    mpay<<' \n' ;
    return output;
}

istream & operator>>(istream & input, sales & s)
{
    cout<<"请依次输入姓名、性别、等级、销售总量:"<<' \n' ;
    input>>s.name>>s.sex>>s.grade>>s.amount;
    s.accumpay=s.pay_calculate();
    return input;
}

void sales::adjust_accumpay()

```

```

{
    cout<<"请输入销售提成比例:";
    double p;
    cin>>p;
    propotion_sales_pre=propotion_sales;
    propotion_sales=p;
}

double sales::get_propotion_S()
{
    return propotion_sales;
}

double sales::get_propotion_Spre()
{
    return propotion_sales_pre;
}

class sales_manager:public manager,public sales
{
public:
    sales_manager();
    sales_manager(char n[],char s[],char g,double a);
    virtual double pay_calculate();
    friend ostream & operator<<(ostream & output,sales_manager &
s);
    friend istream & operator>>(istream & input,sales_manager &
s);

    virtual void adjust_accumpay();
    double static get_propotion_SM();
    double static get_propotion_SMpre();
private:

```

```

        double static propotion_sales_manager;
        double static propotion_sales_manager_pre;
};
double sales_manager::propotion_sales_manager=0.08;
double sales_manager::propotion_sales_manager_pre=0.08;
sales_manager::sales_manager()
{
    NO=num-2;
    num=num-1;
}
sales_manager::sales_manager(char n[], char s[], char g, double
a):manager(n, s, g), sales(n, s, g, a)
{
    strcpy(name, n);
    strcpy(sex, s);
    grade=g;
    NO=num-2;
    num=num-1;
    amount=a;
    accumpay=pay_calculate();
}
double sales_manager::pay_calculate()
{
    double a;
    a=accumpay_manager+amount*propotion_sales_manager;
    return a;
}
ostream & operator<<(ostream & output, sales_manager & s)
{

```

```

        output<<"销售经理
"<<' \t' <<s.name<<' \t' <<s.sex<<' \t' <<s.NO<<' \t' <<s.grade<<' \t' <<s.accu
mpay<<' \n' ;

        return output;
    }

    istream & operator>>(istream & input,sales_manager & s)
    {

        cout<<"请依次输入姓名、性别、等级、销售总量:"<<' \n' ;
        input>>s.name>>s.sex>>s.grade>>s.amount;
        s.accumpay=s.pay_calculate();
        return input;
    }

    void sales_manager::adjust_accumpay()
    {

        cout<<"输入提成比例:";
        double b;
        cin>>b;
        propotion_sales_manager_pre=propotion_sales_manager;
        propotion_sales_manager=b;
    }

    double sales_manager::get_propotion_SM()
    {

        return propotion_sales_manager;
    }

    double sales_manager::get_propotion_SMpre()
    {

        return propotion_sales_manager_pre;
    }
}

```

## “Function.cpp”

```
#include<iostream>
#include<fstream>
#include<string>
#include<graphics.h>
using namespace std;
#include"Position.h"
int main()
{
    void shuru();
    void view();
    void search();
    void xiugai();
    void adjust();
    int i;
    for(i=0;;i++)
    {
        cout<<"-----"<<' \n' ;
        cout<<"|                                |"<<' \n' ;
        cout<<"|      耘含企业人事管理系统      |"<<' \n' ;
        cout<<"|          欢迎使用          |"<<' \n' ;
        cout<<"|                                |"<<' \n' ;
        cout<<"-----"<<' \n' ;
        cout<<"          1. 员工数据录入"<<' \n' ;
        cout<<"          2. 数据库总览"<<' \n' ;
        cout<<"          3. 员工信息查询"<<' \n' ;
        cout<<"          4. 员工数据修改"<<' \n' ;
        cout<<"          5. 薪水调整"<<' \n' ;
```

```

        cout<<"          6. 退出"<<' \n' ;
        cout<<"*****"<<' \n' ;

        int choose;
        cout<<"您想要访问的序号: " <<' \n' ;
        cin>>choose;
        switch(choose)
        {
            case 1:shuru();break;
            case 2:view();break;
            case 3:search();break;
            case 4:xiugai();break;
            case 5:adjust();break;
            case 6:cout<<"谢谢使用, 再见!"<<' \n';exit(0);}
        }
        return 0;
    }

    void shuru()
    {
        ofstream ofile("数据库.txt", ios::app);
        if(!ofile)
        {
            cout<<"打开文件失败"<<' \n' ;
            abort();}
        else
        {
            for(int l=0;;++l)
                {cout<<"您想要输入的对象（CEO、经理、技工、销售、销售经
理）和数量:"<<' \n' ;

                    char a[10];

                    int n;

                    cin>>a>>n;

```



```

if(strcmp(a,"经理")==0)
{
    for(int j=0;j<n;++j)
    {
        manager *m;
        m=new manager;
        cin>>*m;
        ofile<<*m;
        cout<<"录入成功"<<' \n' ;
    }
}

if(strcmp(a,"CEO")==0)
{
    for(int j=0;j<n;++j)
    {
        CEO *c;
        c=new CEO;
        cin>>*c;
        ofile<<*c;
        cout<<"录入成功"<<' \n' ;
    }
}

if(strcmp(a,"技工")==0)
{
    for(int j=0;j<n;++j)
    {
        technician *t;
        t=new technician;
        cin>>*t;
    }
}

```

```

        ofile<<*t;
        cout<<"录入成功"<<' \n' ;
    }
}
if(strcmp(a,"销售")==0)
{
    for(int j=0;j<n;++j)
    {
        sales *s;
        s=new sales;
        cin>>*s;
        ofile<<*s;
        cout<<"录入成功"<<' \n' ;
    }
}
if(strcmp(a,"销售经理")==0)
{
    for(int j=0;j<n;++j)
    {
        sales_manager *s;
        s=new sales_manager;
        cin>>*s;
        ofile<<*s;
        cout<<"录入成功"<<' \n' ;
    }
}
cout<<"是否继续输入 (Y/N) :";
char m;
cin>>m;

```

```

        while(m!='Y' && m!='N')
        {
            cout<<"输入错误，请重新输入 Y/N:";
            cin>>m;
        }
        if(m=='N') break;
    }
}

ofile.close();
}

void view()
{
    char position[100][20];
    char name[100][20];
    char sex[100][10];
    char grade[100][20];
    char no[100][20];
    char accumpay[100][20];
    ifstream ifile("数据库.txt");
    if(!ifile)
        cout<<"打开文件失败"<<'\\n';
    else
    {
        int k;
        for(int l=0;l++)
        {
            ifile.getline(position[l], 100, '\\t');
            ifile.getline(name[l], 100, '\\t');
            ifile.getline(sex[l], 100, '\\t');

```

```

        ifile.getline(no[1], 100, '\t');
        ifile.getline(grade[1], 100, '\t');
        ifile.getline(accumpay[1], 100, '\n');
        if(ifile.peek() == EOF)
            {k=1; break;}
    }

    cout<<"-----"<<' \n' ;
    cout<<"|                                |"<<' \n' ;
    cout<<"|          耘含有限公司员工总览          |"<<' \n' ;
    cout<<"|                                |"<<' \n' ;
    cout<<"-----"<<' \n' ;

    for(int l=0; l<=k; ++l)
    {
        cout<<"职位:"<<position[l]<<' \n' ;
        cout<<"姓名:"<<name[l]<<' \n' ;
        cout<<"性别:"<<sex[l]<<' \n' ;
        cout<<"编号:"<<no[l]<<' \n' ;
        cout<<"等级:"<<grade[l]<<' \n' ;
        cout<<"月薪:"<<accumpay[l]<<' \n' ;
        cout<<' \n' ;
    }
}

void search()
{
    char position[100][20];
    char name[100][20];
    char sex[100][10];
    char grade[100][20];

```

```

char no[100][20];
char accumpay[100][20];
ifstream ifile("数据库.txt");
if(!ifile)
    cout<<"打开文件失败"<<'\\n';
else
{
    for(int l=0;;++l)
    {
        ifile.getline(position[l], 100, '\\t');
        ifile.getline(name[l], 100, '\\t');
        ifile.getline(sex[l], 100, '\\t');
        ifile.getline(no[l], 100, '\\t');
        ifile.getline(grade[l], 100, '\\t');
        ifile.getline(accumpay[l], 100, '\\n');
        if(ifile.peek()==EOF) break;
    }
    for(int o=0;;++o)
    {
        cout<<"请选择查询方式: (A)姓名/工号(B)"<<'\\n';
        char p;
        cin>>p;
        while(p!='B' && p!='A')
        {
            cout<<"输入错误, 请重新输入 A/B:";
            cin>>p;
        }
        if(p=='A')
        {

```

```

        char z[20];
        cout<<"请输入姓名:";
        cin>>z;
        for(int l=0;;++l)
        {
            if(strcmp(name[l], z)==0)
            {
                cout<<"职位:"<<position[l]<<' \n' ;

                cout<<"姓名:"<<name[l]<<' \n' ;
                cout<<"性别:"<<sex[l]<<' \n' ;
                cout<<"编号:"<<no[l]<<' \n' ;
                cout<<"等级:"<<grade[l]<<' \n' ;
                cout<<"月薪:"<<accumpay[l]<<' \n' ;

                break;
            }
        }
    }
    else
    {
        char z[5];
        cout<<"请输入工号";
        cin>>z;
        for(int l=0;;++l)
        {
            if(strcmp(no[l], z)==0)
            {

```

```

        cout<<"职
位:"<<position[1]<<' \n' ;//进行输出

        cout<<"姓名:"<<name[1]<<' \n' ;
        cout<<"性别:"<<sex[1]<<' \n' ;
        cout<<"编号:"<<no[1]<<' \n' ;
        cout<<"等级:"<<grade[1]<<' \n' ;
        cout<<"月
薪:"<<accumpay[1]<<' \n' ;

        break;
    }

}

cout<<"是否继续查询 (Y/N) ";
char l;
cin>>l;
while(l!='Y' &&l!='N')
{
    cout<<"输入错误, 请重新输入 Y/N:";
    cin>>l;
}
if(l=='N') break;
}

}

ifile.close();
}

void xiugai()
{
    char position[100][20];
    char name[100][20];

```

```

char sex[100][10];
char grade[100][20];
char no[100][20];
char accumpay[100][20];
ifstream ifile("数据库.txt");
if(!ifile)
    cout<<"打开文件失败"<<'\\n';
else
{
    for(int l=0;;++l)
    {
        ifile.getline(position[l],100,'\\t');
        ifile.getline(name[l],100,'\\t');
        ifile.getline(sex[l],100,'\\t');
        ifile.getline(no[l],100,'\\t');
        ifile.getline(grade[l],100,'\\t');
        ifile.getline(accumpay[l],100,'\\n');
        if(ifile.peek()==EOF) break;
    }
}
ifile.close();
cout<<"请输入您要修改的编号";
char i[5];
cin>>i;
for(int l=0;;++l)
{
    if(strcmp(no[l],i)==0)
    {
        cout<<"请依次输入职位、姓名、性别、等级、月

```



薪:"<<' \n' ;

```
cin>>position[1]>>name[1]>>sex[1]>>grade[1]>>accumpay[1];
    cout<<"修改成功"<<' \n' ;
    break;
}
}
ofstream ofile("数据库.txt",ios::out);
if(!ofile)
    cout<<"无法打开文件"<<' \n' ;
else
{
    for(int l=0;l<employee::num;++l)
    {

        ofile<<position[1]<<' \t' <<name[1]<<' \t' <<sex[1]<<' \t' <<no[1]<<' \t'
        '<<grade[1]<<' \t' <<accumpay[1]<<' \n' ;
    }
}
ofile.close();
}
void adjust()
{
    enum Pos {C,M,T,S,S_M};
    void adjust_manager();
    void adjust_CEO();
    void adjust_tech();
    void adjust_sales();
    void adjust_sales_m();
}
```

```

void update();

Pos x;

for(int j=0;;j++)
{
    cout<<"您需要修改哪一个职位的薪资(CEO、经理、技工、销售、
销售经理):"<<"\n";

    string p;
    cin>>p;
    if(p=="经理") x=M;
    if(p=="CEO") x=C;
    if(p=="技工") x=T;
    if(p=="销售") x=S;
    if(p=="销售经理") x=S_M;
    switch(x)
    {
        case 0:adjust_CEO();break;
        case 1:adjust_manager();break;
        case 2:adjust_tech();break;
        case 3:adjust_sales();break;
        case 4:adjust_sales_m();break;
    }

    cout<<"是否继续修改薪水(Y/N):";

    char a;
    cin>>a;
    while(a!='Y' && a!='N')
    {
        cout<<"输入错误, 请重新输入:";
        cin>>a;
    }
}

```

```

        if(a==' N' ) break;
    }
    cout<<"是否将修改的薪水调整到数据库中? (Y/N)";
    char a;
    cin>>a;
    while(a!=' Y' &&a!=' N' )
    {
        cout<<"输入错误, 请重新输入:";
        cin>>a;
    }
    if(a==' Y' )    update();
}

void adjust_CEO()
{
    cout<<"您能修改的是 CEO 固定薪水, CEO 原薪水
为:"<<CEO::get_accumpay_c()<<' \n' ;
    CEO *c;
    c=new CEO;
    c->adjust_accumpay();
    CEO::num=CEO::num-1;
    cout<<"修改成功"<<' \n' ;
}

void adjust_manager()
{
    cout<<"您能修改的是经理的固定薪水, 经理原薪水
为:"<<manager::get_accumpay_m()<<' \n' ;
    manager *m;
    m=new manager;
    m->adjust_accumpay();
}

```

```

        manager::num=manager::num-1;
        cout<<"修改成功"<<' \n' ;
    }
void adjust_tech()
{
    cout<<"您能修改的是技工的每小时薪水，原每小时薪水
为:"<<technician::get_accumpay_t()<<' \n' ;
    technician *t;
    t=new technician;
    t->adjust_accumpay();
    technician::num=technician::num-1;
    cout<<"修改成功"<<' \n' ;
}
void adjust_sales()
{
    cout<<"你能修改的是销售的提成比例，原比例
为:"<<sales::get_propotion_S()<<' \n' ;
    sales *s;
    s=new sales;
    s->adjust_accumpay();
    sales::num=sales::num-1;
    cout<<"修改成功"<<' \n' ;
}
void adjust_sales_m()
{
    cout<<"你能修改的是销售经理的提成比例，原比例
为:"<<sales_manager::get_propotion_SM()<<' \n' ;
    sales_manager *s;
    s=new sales_manager;

```

```

        s->adjust_accumpay();
        sales_manager::num=sales_manager::num-1;
        cout<<"修改成功"<<' \n' ;
    }
void update()
{
    char position[100][20];
    char name[100][20];
    char sex[100][10];
    char grade[100][20];
    char no[100][20];
    char accumpay[100][20];
    double calculate_t(char a[]);
    double calculate_s(char a[]);
    double calculate_sm(char a[]);
    ifstream ifile("数据库.txt");
    if(!ifile)
        cout<<"打开文件失败"<<' \n' ;
    else
    {
        int k;
        for(int l=0;;++l)
        {
            ifile.getline(position[l],100,' \t');
            ifile.getline(name[l],100,' \t');
            ifile.getline(sex[l],100,' \t');
            ifile.getline(no[l],100,' \t');
            ifile.getline(grade[l],100,' \t');
            ifile.getline(accumpay[l],100,' \n');

```

```

        if(ifile.peek()==EOF)
            {k=1;break;}
    }
    ofstream ofile("数据库.txt",ios::trunc);
    if(!ofile)
        cout<<"打开文件失败"<<' \n' ;
    else
    {
        for(int l=0;l<=k;++l)
        {

            ofile<<position[l]<<' \t' <<name[l]<<' \t' <<sex[l]<<' \t' <<no[l]<<' \t'
            <<grade[l]<<' \t' ;

                if(strcmp(position[l],"CEO")==0)
            ofile<<CEO::get_accumpay_c()<<' \n' ;

                if(strcmp(position[l],"经理")==0)
            ofile<<manager::get_accumpay_m()<<' \n' ;

                if(strcmp(position[l],"技工")==0)
            ofile<<calculate_t(accumpay[l])<<' \n' ;

                if(strcmp(position[l],"销售")==0)
            ofile<<calculate_s(accumpay[l])<<' \n' ;

                if(strcmp(position[l],"销售经理")==0)
            ofile<<calculate_sm(accumpay[l])<<' \n' ;

        }

    }

    ofile.close();

}

ifile.close();
}

```

```

double calculate_t(char a[])
{
    double m=atoi(a);
    return
m/technician::get_accumpay_tpre()*technician::get_accumpay_t();
}

double calculate_s(char a[])
{
    double m=atoi(a);
    return m/sales::get_propotion_Spre()*sales::get_propotion_S();
}

double calculate_sm(char a[])
{
    double m=atoi(a);
    return (m-
manager::get_accumpay_mpre())/sales_manager::get_propotion_SMpre()*sa
les_manager::get_propotion_SM()+manager::get_accumpay_m();
}

```

## 贪吃蛇大作战：

“Map.h”

```
class Map
{
    public:
        Map(const int, const int);
        void draw();
        void set_food();
        void draw_food();
        int get_food_x();
        int get_food_y();
        int get_Map_length();
        int get_Map_width();
    private:
        int Map_length;
        int Map_width;
        int food_x;
        int food_y;
};

Map::Map(const int length, const int width)
{
    initgraph(length, width);
    Map_length=length;
    Map_width=width;
    draw();
    set_food();
}
```



```

void Map::draw()
{
    cleardevice();
    setfillstyle(EMPTY_FILL, WHITE);
    bar(0, 0, get_Map_length(), get_Map_width());
}

```

```

void Map::set_food()
{
    food_x=rand()%(Map_length-40)+20;
    food_y=rand()%(Map_width-40)+20;
    setfillstyle(EMPTY_FILL, BLACK);
    fillellipse(food_x, food_y, 5, 5);
}

```

```

int Map::get_food_x()
{
    return food_x;
}

```

```

int Map::get_food_y()
{
    return food_y;
}

```

```

int Map::get_Map_length()
{
    return Map_length;
}

```

```
}
```

```
int Map::get_Map_width()
```

```
{
```

```
    return Map_width;
```

```
}
```

```
void Map::draw_food()
```

```
{
```

```
    setfillstyle(EMPTY_FILL, BLACK);
```

```
    fillellipse(food_x, food_y, 5, 5);
```

```
}
```

### **“snake.h”**

```
class snake
```

```
{
```

```
    public:
```

```
        snake(const int, const int, Map *map);
```

```
        void draw();
```

```
        void move_left();
```

```
        void move_right();
```

```
        void move_up();
```

```
        void move_down();
```

```
        void move_forward();
```

```
        int eat_food();
```

```
        bool judge_life();
```

```

    bool get_life();
    char get_head_direction();
    void change_head_direction(char);
    Map *m;
    void change_life(bool);
private:
    int body_x[100];
    int body_y[100];
    int snake_length;
    bool life;
    char head_direction;
};

snake::snake(const int length,const int width,Map *map)
{
    m=map;
    snake_length=3;
    life=true;
    head_direction='s';
    body_x[0]=length/2;
    body_y[0]=width/2;
    for(int i=1;i<snake_length;++i)
    {
        body_x[i]=body_x[i-1];
        body_y[i]=body_y[i-1]-10;
    }
    draw();
}

```

```

void snake::draw()
{
    setfillstyle(EMPTY_FILL, GREEN);
    fillellipse(body_x[0], body_y[0], 5, 5);
    setfillstyle(EMPTY_FILL, BROWN);
    for(int i=1; i<snake_length; ++i)
    {
        fillellipse(body_x[i], body_y[i], 5, 5);
    }
}

```

```

void snake::move_left()
{
    body_x[snake_length]=body_x[snake_length-1];
    body_y[snake_length]=body_y[snake_length-1];
    for(int i=snake_length-1; i>=1; --i)
    {
        body_x[i]=body_x[i-1];
        body_y[i]=body_y[i-1];
    }
    body_x[0]=body_x[0]-10;
    draw();
}

```

```

void snake::move_right()
{
    body_x[snake_length]=body_x[snake_length-1];
    body_y[snake_length]=body_y[snake_length-1];
    for(int i=snake_length-1; i>=1; --i)

```

```

{
    body_x[i]=body_x[i-1];
    body_y[i]=body_y[i-1];
}
body_x[0]=body_x[0]+10;
draw();
}

```

```

void snake::move_up()
{
    body_x[snake_length]=body_x[snake_length-1];
    body_y[snake_length]=body_y[snake_length-1];
    for(int i=snake_length-1;i>=1;--i)
    {
        body_x[i]=body_x[i-1];
        body_y[i]=body_y[i-1];
    }
    body_y[0]=body_y[0]-10;
    draw();
}

```

```

void snake::move_down()
{
    body_x[snake_length]=body_x[snake_length-1];
    body_y[snake_length]=body_y[snake_length-1];
    for(int i=snake_length-1;i>=1;--i)
    {
        body_x[i]=body_x[i-1];
        body_y[i]=body_y[i-1];
    }
}

```

```

    }
    body_y[0]=body_y[0]+10;
    draw();
}

```

```
double sqrt(double);
```

```

int snake::eat_food()
{
    if(sqrt(double(body_x[0]-m->get_food_x())*(body_x[0]-
m->get_food_x())+(body_y[0]-m->get_food_y())*(body_y[0]-
m->get_food_y()))<(double)10)
    {
        snake_length=snake_length+1;
        draw();
        return 1;
    }
    else
        return 0;
}

```

```

bool snake::judge_life()
{
    int k=0;
    for(int i=1;i<snake_length-1;++i)
    {
        if(sqrt(double((body_x[0]-body_x[i])*(body_x[0]-
body_x[i]))+(body_y[0]-body_y[i])*(body_y[0]-body_y[i]))<10)
            k=1;
    }
}

```

```

    }

    if((body_x[0]-0)<5||((m->get_Map_length()-1)-body_x[0])<5
        ||body_y[0]<5||((m->get_Map_width()-1)-body_y[0])<5||k)
        return false;
    else
        return true;

}

bool snake::get_life()
{
    return life;
}

void snake::change_head_direction(char k)
{
    head_direction=k;
}

char snake::get_head_direction()
{
    return head_direction;
}

void snake::move_forward()
{
    if(head_direction=='a')
    {
        for(int i=snake_length-1;i>=1;--i)

```

```

        {
            body_x[i]=body_x[i-1];
            body_y[i]=body_y[i-1];
        }
        body_x[0]=body_x[0]-10;
    }
    if(head_direction=='s')
    {
        for(int i=snake_length-1;i>=1;--i)
        {
            body_x[i]=body_x[i-1];
            body_y[i]=body_y[i-1];
        }
        body_y[0]=body_y[0]+10;
    }
    if(head_direction=='d')
    {
        for(int i=snake_length-1;i>=1;--i)
        {
            body_x[i]=body_x[i-1];
            body_y[i]=body_y[i-1];
        }
        body_x[0]=body_x[0]+10;
    }
    if(head_direction=='w')
    {
        for(int i=snake_length-1;i>=1;--i)
        {
            body_x[i]=body_x[i-1];

```



```

        body_y[i]=body_y[i-1];
    }
    body_y[0]=body_y[0]-10;
}
draw();
}

```

```

void snake::change_life(bool b)
{
    life=b;
}

```

## “Control.h”

```

enum Direction{a, s, d, w, q};

class control
{
public:
    control(snake *S);
    void start(int, int);
    void game_over();
    bool left();
    bool right();
    bool up();
    bool down();
}

```

```

        void show_score();
        void forward();
        int operator +(int);
private:
        char direction;
        Direction D;
        snake *sna;
        int score;
};

control::control(snake *S)
{
    sna=S;
    direction='q';
    score=0;
}

void control::start(int p,int o)
{
    while(1)
    {
        if(sna->get_life())
        {
            if(kbhit())
            {
                direction=getch();
            }

            if(direction!='a' && direction!='s' && direction!='d' && direction!='w'

```

)

```
D=q;
if(direction=='a')
    D=a;
if(direction=='s')
    D=s;
if(direction=='d')
    D=d;
if(direction=='w')
    D=w;
switch (D)
{
    case 0:sna->change_life(left());break;
    case 1:sna->change_life(down());break;
    case 2:sna->change_life(right());break;
    case 3:sna->change_life(up());break;
    case 4:forward();break;
}
if(sna->get_life())
    sna->change_life(sna->judge_life());
if(o)
    Sleep(750-150*p);
else
{
    if(score<=13)
        Sleep(750-50*score);
    else
        Sleep(100);
}
```

```

        cleardevice();
        setfillstyle(EMPTY_FILL, WHITE);

bar(0, 0, sna->m->get_Map_length(), sna->m->get_Map_width());
        if(sna->eat_food())
        {
            score++;
            sna->m->set_food();
        }
        else
            sna->m->draw_food();
    }
else
    {
        game_over();
        break;
    }
}

bool control::left()
{
    if(sna->get_head_direction() != 'd')
    {
        sna->move_left();
        sna->change_head_direction('a');
        return true;
    }
else

```

```

        return false;
    }

bool control::right()
{
    if(sna->get_head_direction()!='a')
    {
        sna->move_right();
        sna->change_head_direction('d');
        return true;
    }
    else
        return false;
}

```

```

bool control::up()
{
    if(sna->get_head_direction()!='s')
    {
        sna->move_up();
        sna->change_head_direction('w');
        return true;
    }
    else
        return false;
}

```

```

bool control::down()
{

```

```

    if(sna->get_head_direction()!='w')
    {
        sna->move_down();
        sna->change_head_direction('s');
        return true;
    }
    else
        return false;
}

```

```

void control::forward()
{
    sna->move_forward();
}

```

```

void control::game_over()
{
    char a[]="DEAD";
    cleardevice();
    setfillstyle(EMPTY_FILL, WHITE);
    bar(0, 0, 500, 500);
    setcolor(RED);
    setfontbkcolor(WHITE);
    setfont(35, 0, "宋体");
    outtextxy(200, 200, a);
    char name[10][20];
    char scor[10][5];
    ifstream ifile("数据库.txt");
    int k;
}

```

```

for(int i=0;;++i)
{
    ifile.getline(name[i],100,'\t');
    ifile.getline(scor[i],100,'\n');
    k=i;
    if(ifile.peek()==EOF) break;
}
ifile.close();
char s[5];
sprintf(s,"%d",score);
strcpy(scor[k],s);
ofstream ofile("数据库.txt",ios::trunc);
for(int i=0;i<=k;++i)
{
    ofile<<name[i]<<'\t'<<scor[i]<<'\n';
}
ofile.close();
}

```

```

void control::show_score()
{
    char a[5];
    sprintf(a,"%d",score);
    setcolor(BLACK);
    setfontbkcolor(WHITE);
    setfont(20,0,"宋体");
    outtextxy(180,235,"您的得分是：");
    outtextxy(300,235,a);
}

```

```

int control::operator+(int k)
{
    return score+k;
}

```

**“Time.h”**

```

class Time
{
    public:
        Time();
        virtual void calculator_time()=0;
    protected:
        int time_all;
};

```

```

Time::Time()
{
    time_all=0;
}

```

```

class time1:public Time
{
    public:
        time1(int, int, control *c);
        virtual void calculator_time();
        int get_time();
}

```



```

private:
    int level;
    int which_kind;
    control *con;
};

timel::timel(int a,int b,control *c)
{
    level=a;
    which_kind=b;
    con=c;
}

void timel::calculator_time()
{
    time_t time_Begin,time_End;
    time_Begin=time(NULL);
    con->start(level,which_kind);
    time_End=time(NULL);
    time_all=time_End-time_Begin;
    con->show_score();
    setfontbkcolor(WHITE);
    setfont(20,0,"宋体");
    outtextxy(180,255,"您所用时间：");
    char t[5];
    sprintf(t,"%d",time_all);
    outtextxy(300,255,t);
    getch();
}

```

**“main.cpp”**

```
#include<iostream>
#include<Windows.h>
#include<graphics.h>
#include<cmath>
#include<fstream>
#include<time.h>
using namespace std;
#include"map.h"
#include"snake.h"
#include"control.h"
#include"Time.h"

int score_all;

int main()
{
    void creat_account();
    void read_rule();
    void greedy_snake1();
    void greedy_snake2();
    void hero_ordor();
    void log_out();
```

```

for(int i=0;;++i)
{
    initgraph(500, 500);
    setfillstyle(EMPTY_FILL, WHITE);
    bar(0, 0, 500, 500);
    setcolor(BLACK);
    setfontbkcolor(WHITE);
    setfont(16, 0, "宋体");
    outtextxy(100, 100, "-----");
    outtextxy(100, 116, " | ");
    outtextxy(100, 132, " | 贪吃蛇大作战 |");
    outtextxy(100, 148, " | ");
    outtextxy(100, 164, "-----");
    outtextxy(100, 180, "*****");
    outtextxy(100, 196, " 1. 创建游戏账号");
    outtextxy(100, 212, " 2. 阅读游戏规则");
    outtextxy(100, 228, " 3. 贪吃蛇 1——匀速");
    outtextxy(100, 244, " 4. 贪吃蛇 2——变速");
    outtextxy(100, 260, " 5. 英雄榜");
    outtextxy(100, 276, " 6. 退出游戏");
    outtextxy(100, 292, "*****");
    outtextxy(100, 308, " 请输入序号");
    setfont(14, 0, "宋体");
    setcolor(RED);
    outtextxy(50, 400, "注：必须先创建账户!!! 创建三个账户之后才
能进行英雄榜功能!! ");
    outtextxy(100, 450, "在进行游戏之前必须创建一个游戏账户");
    char a;
    a=getch();
}

```

```

        switch(a)
        {
            case '1':creat_account();break;
            case '2':read_rule();break;
            case '3':greedy_snake1();break;
            case '4':greedy_snake2();break;
            case '5':hero_ordor();break;
            case '6':log_out();break;
            default:break;
        }
    }
}

void creat_account()
{
    ofstream ofile("数据库.txt", ios::app);
    char a[20];
    int i;
    cleardevice();
    setfillstyle(EMPTY_FILL, WHITE);
    bar(0, 0, 500, 500);
    setcolor(BLACK);
    setfontbkcolor(WHITE);
    setfont(16, 0, "宋体");
    inputbox_getline("请输入您的名字: ", "请输入您的游戏昵称(回车确
认)", a, 20);
    ofile<<a<<' \t' <<"0"<<' \n' ;
    ofile.close();
}

```

```

void read_rule()
{
    cleardevice();
    setfillstyle(EMPTY_FILL, WHITE);
    bar(0, 0, 500, 500);
    setcolor(BLACK);
    setfontbkcolor(WHITE);
    setfont(14, 0, "宋体");
    outtextxy(80, 100, "游戏规则：");
    outtextxy(80, 116, "1. 贪吃蛇控制：a(左), s(下), d(右), w(上)");
    outtextxy(80, 132, "2. 当贪吃蛇碰到自己身体或者边界的时候，死亡");
    outtextxy(80, 148, "3. 选择匀速时，等级从 1-4, 速度呈梯度上升");
    outtextxy(80, 164, "4. 选择变速，得分越多，速度越快，当速度足够快的时
候，会匀速");
    outtextxy(80, 180, "5. 绿色为蛇头，棕色为蛇身");
    outtextxy(80, 196, "6. 初始方向竖直向下");
    outtextxy(80, 212, "按任意键回到菜单");
    char a;
    a=getch();
}

```

```

void greedy_snake1()
{
    int length=500;
    int width=500;
    int i=0;
    cleardevice();
    setfillstyle(EMPTY_FILL, WHITE);

```

```

bar(0, 0, 500, 500);
setcolor(BLACK);
setfontbkcolor(WHITE);
setfont(20, 0, "宋体");
outtextxy(170, 200, "请选择难度");
outtextxy(170, 220, "1. 简单");
outtextxy(170, 240, "2. 一般");
outtextxy(170, 260, "3. 困难");
outtextxy(170, 280, "4. 绝望");
char a;
a=getch();
switch(a)
{
    case '1':i=1;break;
    case '2':i=2;break;
    case '3':i=3;break;
    case '4':i=4;break;
}
Map m(length,width);
snake s(length,width,&m);
control c(&s);
time1 t(i, 1, &c);
getch();
t.calculator_time();
score_all=c+score_all;
}

void greedy_snake2()
{

```

```

    int length=500;
    int width=500;
    Map m(length,width);
    snake s(length,width,&m);
    control c(&s);
    timer t(0,0,&c);
    getch();
    t.calculator_time();
    score_all=c+score_all;
}

void hero_order()
{
    char name[10][20];
    char score[10][5];
    int sc[10];
    ifstream ifile("数据库.txt");
    int k;
    for(int i=0;;++i)
    {
        ifile.getline(name[i],100,'\t');
        ifile.getline(score[i],100,'\n');
        k=i;
        if(ifile.peek()==EOF) break;
    }
    ifile.close();
    for(int i=0;i<=k;++i)
    {
        sc[i]=atoi(score[i]);
    }
}

```

```

}
for(int i=0;i<=k;++i)
{
    for(int j=i+1;j<=k;++j)
    {
        if(sc[j]>sc[i])
        {
            int t;
            char n[20];
            t=sc[j];sc[j]=sc[i];sc[i]=t;
            strcpy(n,name[i]);
            strcpy(name[i],name[j]);
            strcpy(name[j],n);
        }
    }
}

char a[3][5];
sprintf(a[0],"%d",sc[0]);
sprintf(a[1],"%d",sc[1]);
sprintf(a[2],"%d",sc[2]);
cleardevice();
setfillstyle(EMPTY_FILL,WHITE);
bar(0,0,500,500);
setcolor(BLACK);
setfontbkcolor(WHITE);
setfont(20,0,"宋体");
outtextxy(150,100,"英雄榜");
outtextxy(150,125,"第一名：");
outtextxy(230,125,name[0]);

```



```

    outtextxy(300, 125, a[0]);
    outtextxy(150, 150, "第二名: ");
    outtextxy(230, 150, name[1]);
    outtextxy(300, 150, a[1]);
    outtextxy(150, 175, "第三名: ");
    outtextxy(230, 175, name[2]);
    outtextxy(300, 175, a[2]);
    outtextxy(150, 200, "所有玩家总得分: ");
    char sco[5];
    sprintf(sco, "%d", score_all);
    outtextxy(350, 200, sco);
    getch();
}

void log_out()
{
    cleardevice();
    setfillstyle(EMPTY_FILL, WHITE);
    bar(0, 0, 500, 500);
    setcolor(BLACK);
    setfontbkcolor(WHITE);
    setfont(30, 0, "宋体");
    outtextxy(130, 100, " 谢 谢 使 用!");
    setfont(15, 0, "宋体");
    outtextxy(130, 130, "*****");
    outtextxy(130, 145, "      出品人: 漆耘含");
    outtextxy(130, 160, "      监制: 漆耘含");
    outtextxy(130, 175, "      创作时间: 2017. 7. 2");
    getch();
}

```

```
    exit(0);  
}
```

## 进度安排：

（这是在小学期前两周的安排，在小学期开始之前就写完了）

小学期 第一周 (人事管理系统)	周一	编写类
	周二	编写类
	周三	编写数据录入 功能
	周四	编写数据总览和 个人信息查询功 能
	周五	编写个人信息修 改和基础薪水修 改模块
小学期 第二周 (贪吃蛇)	周一	编写类
	周二	编写类
	周三	实现贪吃蛇匀速 前进功能
	周四	实现贪吃蛇变速 前进功能
	周五	实现创建账户功 能和英雄榜功能

## 评分表：

第一题：

项目	评价	
设计方案的合理性与创新性	3	
设计与调试结果	4	
设计说明书的质量	1	
程序基本要求涵盖情况	4	
程序代码编写素养情况	2	
课程设计周表现情况	1	
综合成绩	15	

第二题：

项目	评价	
选题报告与设计说明说	1	
程序基本要求涵盖情况	4	
程序扩展要求与创新	3	
程序代码编写素养情况	2	
设计与运行结果	4	
综合成绩	15	

教师签名：\_\_\_\_\_

日期：\_\_\_\_\_