# 课程设计报告书 1803粟锦

* 1. 选题

聪明的小蛇

* 1. 需求分析

1. 有多种东西让小蛇吃。
2. 根据用户的需求设置游戏的不同难度（游戏中将这些设置成为用户自己定义各种东西的数量以及移动速度）
3. 有不同的关卡供用户在通关后可以自由选择已通关的关卡。
4. 用户可以自行DIY游戏地图。
5. 增加了背景音乐（用户可自行关闭）。
6. 建立了用户登录系统，用户需要创建自己的账号才能登录。
7. 有保存进度功能，可以让用户继续游戏。
8. 设置了排行榜功能，各个用户共享排行榜上的分数。
9. 界面美观，赏心悦目。
   1. 系统设计

1.游戏进行：将数组与链表相结合，链表记录蛇每一节的坐标，二维数组记录每一个移动方格中蕴含的值（不同的值代表方格中有着不同的东西）。蛇的移动始终与方格的大小对应。因此可以通过判断蛇头所在方格里的值来确定蛇吃到了什么东西。

结构：char map[54][29] ,全局变量地图数组。

其中0为通路，1为食物，2为毒草，3为障碍物, 4为地雷,5为蛇自己, 6为传送门, 7为下一个入口；

struct snake //定义蛇结构体

{

int x;

int y;

struct snake \* previous;

struct snake \* next;

}

struct snake \* head, \*tail 蛇的头尾指针

2.读取地图与进度：运用文件输入输出，将地图文件以及用户正在进行的游戏保存下来，让用户可以继续游戏并根据闯关进度选择关卡。其中地图文件是将关卡中数组中的值记录在文件里，在打开后读取里面的值并赋给相应数组中的位置。读取进度则增添了蛇的长度，方向，各个节点的坐标位置以及关卡的选择

结构：

Char sign; FILE fp;

Fopen\_s(&fp,address,”r”); //读取地图文件

for (int i = 1; i < 53; i++)

for (int j = 1; j < 28; j++)

{

sign = fgetc(fp) - '0';

map[i][j] = sign;

}

1. 背景音乐： playsound函数。

4.排行榜：读取文件中前10行数字，然后加上用户新创建的函数，进行一次选择排序，将前10个文件重新输入文件中。

5.用户登录系统：

（1）创建新用户：人工制造用户名输入框与密码输入框，用\_getch()获取用户输入的字符并模拟在输入框输入的效果。用户确认后进行判断：如果能够在指定目录下打开以用户名为名称的dat文件，则根据fopen\_s()的返回值提示用户该用户名已存在。否则建立一个以用户名为名称的dat文件，并将密码保存在文件的第一行。第二行输入用户通过的关卡数（初始为0）。进入用户主界面。

1. 用户登录：根据用户输入的用户名，利用fopen\_s()来判断该用户名是否存在，如若存在则读取该用户名文件的第一行密码，与用户输入的密码进行对比，如果完全相同则显示登录成功，进入用户主界面，否则显示密码输入错误。

结构： char filename[20], real\_password[20];（存储用户名与真实密码）

char address[100];（存储打开路径）

char dat[6] = ".dat";

char txt[6] = ".txt";

Load\_address() ：自定义函数来将用户输入的用户名与默认路径相结合形成要打开的用户名文件路径。

1. 用户DIY地图：

通过特定按键是用户可以在数组的特定位置生成障碍物（即将该位置的元素赋值为障碍物）。当用户确定完后，利用文件输入将数组中的元素全部打印进地图文件里，并在开始游戏时进行调用。

读取结构：Char sign; FILE fp;

Fopen\_s(&fp,address,”r”); //输入地图文件

for (int i = 1; i < 53; i++)

for (int j = 1; j < 28; j++)

{

sign = map[i][j] + '0';

fputc(sign, fp);

}

7.界面美观：运用图形界面以及多方面贴图，尽量达到界面令人赏心悦目的程度。

* 1. 界面设计

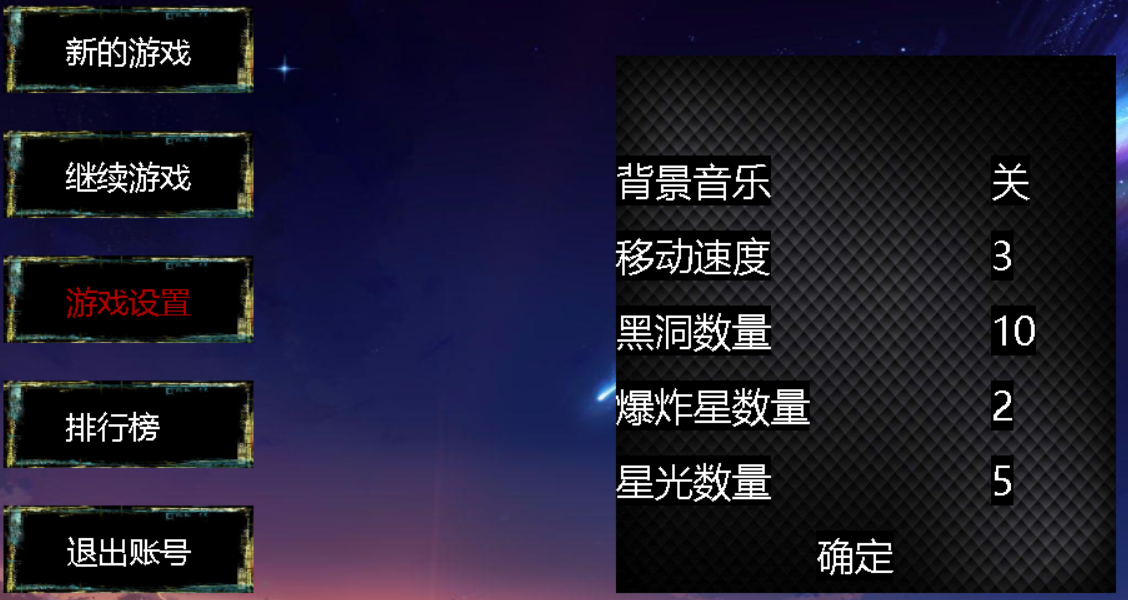
用户登录前的主界面：



用户登录后的主界面：



将不同的选项分别以弹框形式提供用户使用：





* 1. 系统实现

系统主体是循环，通过WHILE两个循环实现游戏的重复进行：

void choose()

{

int i, j, n, judge, judge0 ;

while (1) //第一个循环是主界面的循环，即用户未登录前的界面的循环。

{ // i代表用户确定了主界面的第几个选项

judge = 1;

i = mainmenu();

switch (i)

{

case 1:;

case 2: if (i == 2 && keepgame() != 0 || i==1 && client() != 0)

while (1) //第二个循环是用户登录后的循环，即用户界面。

{ //n代表了用户确定了用户界面的第几个选项

n = client\_menu();

judge0 = 1;

j = 1;

switch (n)

{

case 2:

if(KeepOnGame() == 0)

break;

case 1: if(n != 2)

j = game\_choose();

while (judge0 == 1 && j != 0)//循环代表用户输掉后是否重新游戏

{

judge0 = game\_start(); //正式开始一局游戏

} break;

case 3: set(); break;

case 4: RANK(); break;

case 5: judge0 = -1;break;

}

if (judge0 == -1) //用户退出账户，回到主界面

break;

}break;

case 3: RANK(); break;

case 4: judge = -1;break;

}

if (judge == -1) //用户退出游戏，游戏关闭

break;

}

}

游戏开始程序：

int game\_start()

{

int i;

key = RIGHT; //代表小蛇初始方向向右

snake\_num = 5; //代表小蛇的初始长度

graph(); //读取地图文件，并定义食物毒草之类的东西

print(); //遍历数组，将数组元素值代表的东西打印出来

print\_snake(); //通过链表单独进行蛇的打印

i = move(); //主要函数

return i; //返回值代表用户的选择（重新游戏或返回用户界面）

}

move(） 函数，以伪代码形式表达

{

蛇移动前的游戏提示环节；

While(1)

{

获取用户的键盘相应并作出相应的判断；

对蛇头坐标的数组元素值进行判断来判断蛇头吃到了什么东西并进行相应回复；

If(蛇死亡）

Break;

}

跳出弹框提示用户输了；

将用户所得分数进行排行操作，得出新的排行榜；

While(1)

{

用户进行选择（重新游戏或返回主界面）

选择后return ,根据返回值判断下一步

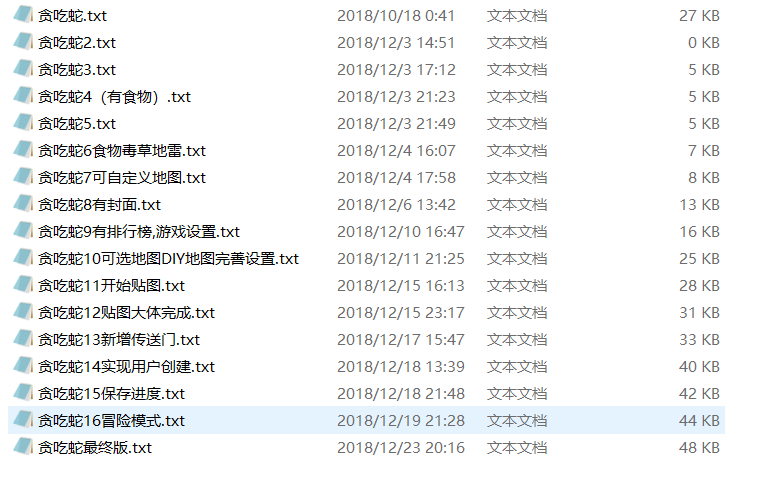
}

清除蛇链表；

}

* 1. 测试与调试

程序的制作一共有以下若干版本：



* 1. 心得与体会

这次课设我真的受益匪浅。想起老师曾说过的：“做一次这样的课设可以很大程度地提高自己写代码的能力。”这句话我如今感同身受，因为做这样一个课设非常的考研自己对语言整体的掌握能力，各个方面都要有所涉猎。同时自己也在制作过程中遭遇了各式各样的BUG，大大地丰富了我的经验。我能清楚地感觉到自己写代码的能力得到了很大的提升！