课程设计报告书

一、选题

聪明的小蛇

二、需求分析

程序中必须要有地图，能移动的小蛇，会闪烁的毒草，会自动刷新的食物以及吃到食物就变色，地雷，三个关卡，排行榜。每个关卡设置不同的地图，因此要设置一个状态量chapterstatus==1，2，3,其值代表对应的关卡1，2，3。对于蛇本身采用指针的数据结构（单向指针），吃到食物加一截长度，吃到毒草减一截长度，吃到地雷长度减半。关卡之前可以由低到高，由分数值来进行条件判断。蛇的移动方向采用上下左右的方向键实现。从不同吃到方向吃到食物会变色，要在方向的条件判断里加上变色功能。

三、系统设计

1. 数据结构设计：列出关键的数据结构；

蛇的方向键设置

#define U 1

#define D 2

#define L 3

#define R 4

//蛇的状态，U：上 ；D：下；L:左 R：右

蛇的数据结构（单向链表）

typedef struct SNAKE //蛇身的一个节点

{

int x;

int y;

struct SNAKE \*next;

}snake;

食物，毒草，地雷——结构体

struct LIST

{

int x;

int y;

}food,poison,boom;

//全局变量//

int n = 1;

int count = 51;

int ducao = 6;

int score = 0, add = 10;//总得分与每次吃食物得分。

int status, sleeptime = 200;//每次运行的时间间隔

snake \*head;

struct LIST food;

struct LIST boom;

struct LIST poison1;

struct LIST poison2;

struct LIST poison3;

snake \*q;//遍历蛇的时候用到的指针

int endgamestatus = 0; //游戏结束的情况，1：撞到墙；2：咬到自己；3：主动退出游戏；4：被炸死了；5:毒草

int chapterstatus;

int top\_score[5];//记录和比较分数的数组

1. 算法和程序流程：画出相应程序流程图；

手写

1. 模块间接口描述

int main()

{

PlaySoundA ("bgm.wav", NULL, SND\_FILENAME | SND\_ASYNC | SND\_LOOP);

gamestart();

gamecircle();

return 0;

}//main函数由gamestart和gamecircle两个函数组成

void gamestart()//游戏初始化

{

system("mode con cols=100 lines=30");

welcometogame();

chapter();//chapter里的chapterstatus和creatMap里的关卡地图对应

creatMap();

initsnake();

createfood\_1();

createboom();

createpoison();

}

void snakemove()里包含biteself()和cantcrosswall（）这两个函数里由包含着endgame（）函数，同时snakemove里由createfood（）creadtepoison() createboom()函数，吃完一个之后可以出现下一个食物，毒草，地雷

endgame（）函数里可以有gamestart与gamecricle，使得游戏可以继续玩。

四、界面设计

手写

五、系统实现

int n = 1;

int count = 51;

int ducao = 6;

int score = 0, add = 10;//总得分与每次吃食物得分。

int status, sleeptime = 200;//每次运行的时间间隔

snake \*head;

struct LIST food;

struct LIST boom;

struct LIST poison1;

struct LIST poison2;

struct LIST poison3;

snake \*q;//遍历蛇的时候用到的指针

int endgamestatus = 0; //游戏结束的情况，1：撞到墙；2：咬到自己；3：主动退出游戏；4：被炸死了；5:毒草

int chapterstatus;

int top\_score[5];//记录和比较分数的数组

//声明全部函数//

void Pos();

void BubbleSort();

void SaveRecord();

void LoadRecord();

void creatMap();

void initsnake();

int biteself();

void createfood\_1();

void createboom();

void createpoison();

void cantcrosswall();

void snakemove();

void pause();

void check();//防止刷新时重合

void gamecircle();

void welcometogame();

void endgame();

void gamestart();

void chapter();//关卡选择

void snakemove()//蛇前进,上U,下D,左L,右R

{

snake \* nexthead;

cantcrosswall();

nexthead = (snake\*)malloc(sizeof(snake));

if (status == U)

{

nexthead->x = head->x;

nexthead->y = head->y - 1;

}

if (status == D)

{

nexthead->x = head->x;

nexthead->y = head->y + 1;

}

if (status == L)

{

nexthead->x = head->x - 2;

nexthead->y = head->y;

}

if (status == R)

{

nexthead->x = head->x + 2;

nexthead->y = head->y;

}

if (head->x == food.x && head->y == food.y)//如果下一个有食物//

{

if(status==U)

system("color C");

if(status==D)

system("color A");

if(status==L)

system("color D");

if(status==R)

system("color E");

nexthead->next = head;

head = nexthead;

q = head;

while (q != NULL)

{

Pos(q->x, q->y);

printf("■");

q = q->next;

}

score = score + add;

createfood\_1();

check();

}

else if (head->x == boom.x&&head->y == boom.y)//如果下一个有炸弹

{

q = head;

n = 1;

while (q->next != NULL)

{

n++;

q = q->next;

}

if (n <= 2)

{

endgamestatus = 4;

endgame();

}

if (n % 2 == 0)

n = n / 2;

else

n = (n - 1) / 2;

for (int i = 0; i<n; i++)//擦掉蛇身一半

{

q = head;

while (q->next->next != NULL)

{

q = q->next;

}

Pos(q->next->x, q->next->y);

printf(" ");

q->next = NULL;

}

score = score - add / 2;

createboom();

check();

}

else if ((head->x == poison1.x&&head->y == poison1.y)||(head->x == poison2.x&&head->y == poison2.y)||(head->x == poison3.x&&head->y == poison3.y))//如果下一个有毒草

{

n = 1;

q = head;

while (q->next != NULL)

{

n++;

q = q->next;

}

if (n <= 2)

{

endgamestatus = 5;

endgame();

}

q = head;

while (q->next->next != NULL)

{

q = q->next;

}

Pos(q->next->x, q->next->y);

printf(" ");

free(q->next);

q->next = NULL;

score = score - add / 2;

Pos(poison1.x,poison1.y);

printf(" ");

Pos(poison2.x,poison2.y);

printf(" ");

Pos(poison3.x,poison3.y);

printf(" ");

createpoison();

check();

}

else //如果什么都没有//

{

count++;

ducao++;

nexthead->next = head;

head = nexthead;

q = head;

while (q->next->next != NULL)

{

Pos(q->x, q->y);

printf("■");

q = q->next;

}

Pos(q->next->x, q->next->y);

printf(" ");

q->next = NULL;

}

if (biteself() == 1) //判断是否会咬到自己

{

endgamestatus = 2;

endgame();

}

}

void check()

{

if ((food.x == boom.x&&food.y == boom.y) || (food.x == poison1.x&&food.y == poison1.y)|| (food.x == poison2.x&&food.y == poison2.y)|| (food.x == poison3.x&&food.y == poison3.y))

{

createfood\_1();

}

if ((boom.x == poison1.x&&boom.y == poison1.y)||(boom.x == poison2.x&&boom.y == poison2.y)||(boom.x == poison3.x&&boom.y == poison3.y))

{

createboom();

}

}

void pause()//暂停

{

while (1)

{

Sleep(300);

if (GetAsyncKeyState(VK\_SPACE))

{

break;

}

}

}

void chapter()//关卡选择界面

{

Pos(40, 12);

printf("1.无聊的第一关。\n");

Pos(40, 16);

printf("2.有点改变的第二关。\n");

Pos(40, 20);

printf("3.还可以玩玩的第三关。\n");

Pos(40, 24);

printf("4.没啥用的排行榜。\n");

Pos(40, 28);

printf("你的选择是：");

scanf("%d", &chapterstatus);

if (chapterstatus > 4)

{

system("cls");

printf("抱歉，你不能选择不存在的关卡。\n");

system("pause");

system("cls");

chapter();

}

else if (chapterstatus == 4)

{

system("cls");

LoadRecord();

system("pause");

system("cls");

chapter();

}

else

system("cls");

}

void gamecircle()//控制游戏

{

Pos(70, 15);

printf("不能穿墙，不能咬到自己\n");

Pos(70, 17);

printf("用↑.↓.←.→分别控制蛇的移动.");

Pos(70, 19);

printf("右回车键 为加速\n");

Pos(70, 20);

printf("右shift键 为减速\n");

Pos(70, 21);

printf("ESC ：退出游戏\n");

Pos(70, 22);

printf("space：暂停游戏.");

status = R;

while (1)

{

if (count % 50 == 0)

{

Pos(food.x, food.y);

printf(" ");

createfood\_1();

check();

}

if (ducao % 7 == 0)

{

Pos(poison1.x, poison1.y);

printf(" ");

Pos(poison2.x, poison2.y);

printf(" ");

Pos(poison3.x, poison3.y);

printf(" ");

}

if (ducao % 9 == 0)

{

Pos(poison1.x, poison1.y);

printf("★");

Pos(poison2.x, poison2.y);

printf("★");

Pos(poison3.x, poison3.y);

printf("★");

}

Pos(70, 10);

printf("得分：%d ", score);

Pos(70, 11);

printf("每个食物得分：%d分", add);

if (GetAsyncKeyState(VK\_UP) && status != D)

{

status = U;

}

else if (GetAsyncKeyState(VK\_DOWN) && status != U)

{

status = D;

}

else if (GetAsyncKeyState(VK\_LEFT) && status != R)

{

status = L;

}

else if (GetAsyncKeyState(VK\_RIGHT) && status != L)

{

status = R;

}

else if (GetAsyncKeyState(VK\_SPACE))

{

pause();

}

else if (GetAsyncKeyState(VK\_ESCAPE))

{

endgamestatus = 3;

break;

}

else if (GetAsyncKeyState(VK\_RETURN))

{

if (sleeptime >= 50)

{

sleeptime = sleeptime - 30;

add = add + 2;

if (sleeptime == 320)

{

add = 2;//防止减到1之后再加回来有错

}

}

}

else if (GetAsyncKeyState(VK\_RSHIFT))

{

if (sleeptime<350)

{

sleeptime = sleeptime + 30;

add = add - 2;

if (sleeptime == 350)

{

add = 1; //保证最低分为1

}

}

}

Sleep(sleeptime);

snakemove();

if (chapterstatus == 1)

{

if (score >= 50 && score <= 100)

{

chapterstatus = 2;

system("cls");

printf("恭喜你通过第一关\n");

system("pause");

system("cls");

creatMap();

initsnake();

createfood\_1();

createboom();

createpoison();

gamecircle();

}

}

if (chapterstatus == 2)

{

if (score>100)

{

chapterstatus = 3;

system("cls");

printf("恭喜你通过第二关\n");

system("pause");

system("cls");

creatMap();

initsnake();

createfood\_1();

createboom();

createpoison();

gamecircle();

}

}

}

}

void welcometogame()//开始界面

{

Pos(40, 12);

system("title 吕义的c语言课设");

Pos(40, 15);

printf("欢迎来到聪明的小蛇游戏");

Pos(40, 25);

system("pause");

system("cls");

}

void endgame()//结束游戏

{

system("cls");

Pos(24, 12);

if (endgamestatus == 1)

{

printf("对不起，您撞到墙了。游戏结束.");

}

else if (endgamestatus == 2)

{

printf("对不起，您咬到自己了。游戏结束.");

}

else if (endgamestatus == 3)

{

printf("您的已经结束了游戏。");

}

else if (endgamestatus == 4)

{

printf("对不起，您被炸死了");

}

else if (endgamestatus == 5)

{

printf("对不起，您被毒毒毒毒毒死了");

}

Pos(24, 13);

printf("您的得分是%d\n", score);

SaveRecord(score);

score = 0;

system("pause");

system("cls");

gamestart();

gamecircle();

}

void gamestart()//游戏初始化

{

system("mode con cols=100 lines=30");

welcometogame();

chapter();

creatMap();

initsnake();

createfood\_1();

createboom();

createpoison();

}

int main()

{

PlaySoundA ("bgm.wav", NULL, SND\_FILENAME | SND\_ASYNC | SND\_LOOP);

gamestart();

gamecircle();

return 0;

}

六、测试与调试

记录测试，调试和程序修改的过程。

void endgame()//结束游戏

{

。。。。。。。

printf("您的得分是%d\n", score);

SaveRecord(score);

score = 0;

system("pause");

system("cls");

gamestart();//实现游戏在结束后能重新开始

gamecircle();

}

加在gamecricle（）里的代码，可以进入下一个关卡

if (score >= 50 && score <= 100)

{

chapterstatus = 2;

system("cls");

printf("恭喜你通过第一关\n");

system("pause");

system("cls");

creatMap();

initsnake();

createfood\_1();

createboom();

createpoison();

gamecircle();

}

if (score>100)

{

chapterstatus = 3;

system("cls");

printf("恭喜你通过第二关\n");

system("pause");

system("cls");

creatMap();

initsnake();

createfood\_1();

createboom();

createpoison();

gamecircle();

endgame();

}

实现毒草的闪烁和食物的刷新

if (count % 50 == 0)

{

Pos(food.x, food.y);

printf(" ");

createfood\_1();

check();

}

if (ducao % 7 == 0)

{

Pos(poison1.x, poison1.y);

printf(" ");

Pos(poison2.x, poison2.y);

printf(" ");

Pos(poison3.x, poison3.y);

printf(" ");

}

if (ducao % 9 == 0)

{

Pos(poison1.x, poison1.y);

printf("★");

Pos(poison2.x, poison2.y);

printf("★");

Pos(poison3.x, poison3.y);

printf("★");

}

将食物的数据类型改成结构体并解决了由多次递归带来的错误

改之前：

void createfood\_1()//随机出现食物

{

snake \*food\_1,\*food\_2;

srand((unsigned)time(NULL));

food\_1=(snake\*)malloc(sizeof(snake));

food\_1->x=rand()%52+2;

while((food\_1->x%2)!=0) //保证其为偶数，使得食物能与蛇头对其

{

food\_1->x=food\_1->x+1;

}

food\_1->y=rand()%24+1;

q=head;

while(q->next!=NULL)

{

if(q->x==food\_1->x && q->y==food\_1->y) //判断蛇身是否与食物重合

{

free(food\_1);

createfood\_1();

}

q=q->next;

}

Pos(food\_1->x,food\_1->y);

food=food\_1;

printf("●");

}

改之后：

void createfood\_1()//随机出现食物

{

int i;

srand((unsigned)time(NULL));

loop: food.x = rand() % 56 + 2;

while ((food.x % 2) != 0) //保证其为偶数，使得食物能与蛇头对其

{

food.x = food.x + 1;

}

food.y = rand() % 28 + 1;

q = head;

while (q!= NULL)

{

if (q->x == food.x && q->y == food.y) //判断蛇身是否与食物重合

{

goto loop;

}

q = q->next;

}

写入文件和读取文件的fopen 函数里的w,r,a,a+要注意区别

if(score1>top\_score[4]){

top\_score[4]=score1;

BubbleSort(top\_score,5);

record=fopen("Record.txt","w");

for(i=0;i<5;i++)

fprintf(record,"%d ",top\_score[i]);

fclose(record);

此处的w不能写成a，a+

七、心得与体会

这次的课设可谓是让我痛并快乐着，看着自己的程序从无到现在花样多多的贪吃蛇，这份成就是让我感到非常开心的，然而在编程的过程中却遇到了许多bug，有些还好解决，只是逻辑上的漏洞，然而有的bug却是在逻辑上看起来没有错误的，像我的代码里一开始写到食物是用的递归过多，导致系统报错，在将食物的递归改为goto循环语句后，这个问题便得到了解决。此外，在完成这次课设的任务时，除了学到了编程的知识之外，我觉得还增强了自我学习能力，遇到不懂得的问题，就去找，一步步地把它弄懂，慢慢得积累，学到的东西自然会越来越多。