**中 国 矿 业 大 学**

**级《数据结构与算法分析》课程设计**

学生姓名

学 号

**中国矿业大学信控学院**

**贪吃蛇**

1. **需求分析**

通过在控制台中显示蛇和食物，以及蛇运动的范围(构建围墙)。还包括蛇的移动方法、吃食物增长蛇身的方法、食物随机生成的方法以及速度的控制。接着是进行游戏的判断，判断蛇是否接触墙壁或者身体交叉。以及游戏结束后，是否重新开始游戏，或者退出游戏。为了提高用户体验，还需要实现游戏的暂停和继续。具体包括如下：

1. 打开程序，出现游戏界面；
2. 开始游戏，通过键盘的上下左右键来控制蛇的移动方向；
3. 贪吃蛇撞到墙壁或者自己的身体，游戏结束；
4. 贪吃蛇吃到一个食物后，身体的长度加一，并且游戏分数加一。
5. **问题描述**

贪吃蛇是一个经典的小游戏，一条蛇在封闭的围墙里，围墙中随机出现事务，通过键盘的上下左右键来控制蛇运动的方向，蛇头撞到食物，则食物被吃掉，设身体长一节，同时游戏分数加一，接着下一个食物随机出现，等着蛇过去吃，如果蛇在移动过程撞到墙或者身体交叉，则游戏结束。

1. **算法设计**

在贪吃蛇游戏中，首先初始化界面，包括按键的基本信息、当前分数、一条蛇。然后随机产生一个食物，食物的坐标是随机的，开始被标记为未被吃掉。此时，用户可以通过控制键盘来移动蛇，还可以设置蛇运动的速度，以及游戏的暂停和继续。当蛇吃掉当前食物时，将当前食物标记未被吃掉，并在屏幕上删除，还需要将蛇的长度加1，以及将当前的分数加1。同时当一个食物被吃掉，还需要随机生成下一个食物，以供蛇去吃。如果在移动的过程中，蛇碰到了墙壁，或者蛇的头碰到了蛇的身体，则游戏结束，让用户选择释放继续游戏。如果继续游戏，则回到开头，重新初始化界面。否则，直接退出游戏。具体的流程如下图1所示。



图1 贪吃蛇游戏流程

在编程过程中，使用到的结构体：

蛇的结构体定义蛇的属性：

struct snake {

int len;//当前蛇的长度

struct coor coord[SNAKE\_LENGTH];//每一节蛇的坐标

int key;//定义蛇的方向

int speed;

int flag;//定义蛇的状态 1表示存活 0表示死亡

}snake;

食物结构体描述食物的位置和食物的状态：

struct foodcoord {

int x;

int y;

int flag;//定义食物的状态

}food;

蛇的坐标信息结构体：

struct coor {

int x;

int y;

};

具体的模块：

初始化游戏界面：void init\_sence();

关键移动模块：void gotoxy(int x, int y);

蛇的移动模块：void move\_snake();

打印蛇模块：void draw\_snake();

食物产生模块：void creatfood();

判断蛇是否吃到食物模块：void eatfood();

判断蛇是否死掉模块：void SnakeState();

在我们的程序，通过键盘来控制游戏。其中F1是降低蛇的运动速度，F2是加速蛇的运动速度；空格键是暂停游戏，Ctrl是继续游戏，Esc是退出游戏；上下左右键来控制蛇的运动方向。

1. **源代码**

#include<stdio.h>

#include<stdlib.h>

#include<windows.h>

#include<time.h>

#define SNAKE\_LENGTH 100//定义蛇的最大长度

#define SCREEN\_WIDETH 80

#define SCREEN\_HEIGHT 30

//枚举方向

enum KEY {

right = VK\_RIGHT,

left = VK\_LEFT,

up = VK\_UP,

down = VK\_DOWN

};

//定义每一节蛇的坐标

struct coor {

int x;

int y;

};

//定义蛇的属性

struct snake {

int len;//当前蛇的长度

struct coor coord[SNAKE\_LENGTH];//每一节蛇的坐标

int key;//定义蛇的方向

int speed;

int flag;//定义蛇的状态 1表示存活 0表示死亡

}snake;

//光标移动函数

void gotoxy(int x, int y)

{

COORD pos;

pos.X = x;

pos.Y = y;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), pos);

}

//食物的信息

struct foodcoord {

int x;

int y;

int flag;//定义食物的状态

}food;

//蛇的移动

void move\_snake();

//画出蛇

void draw\_snake();

//产生食物

void creatfood();

//判断蛇是否吃到食物

void eatfood();

//判断蛇是否死掉

void SnakeState();

//初始化游戏界面

void init\_sence()

{

//初始化上下墙

for (int i = 0; i < SCREEN\_WIDETH; i += 2)

{

gotoxy(i, 0);

printf("■");

gotoxy(i, SCREEN\_HEIGHT);

printf("■");

}

//初始化左右墙

for (int i = 0; i <= SCREEN\_HEIGHT; i++)

{

gotoxy(0, i);

printf("■");

gotoxy(SCREEN\_WIDETH, i);

printf("■");

}

//打印提示信息

gotoxy(SCREEN\_WIDETH + 5, 2);

printf("\t贪吃蛇");

gotoxy(SCREEN\_WIDETH + 5, 4);

printf("\t2020-01-07");

gotoxy(SCREEN\_WIDETH + 5, 8);

gotoxy(SCREEN\_WIDETH + 5, 10);

printf("\tF1:加速\n");

gotoxy(SCREEN\_WIDETH + 5, 12);

printf("\tF2：减速\n");

gotoxy(SCREEN\_WIDETH + 5, 14);

printf("\tCTRL:继续\n");

gotoxy(SCREEN\_WIDETH + 5, 16);

printf("\t空格：暂停\n");

gotoxy(SCREEN\_WIDETH + 5, 18);

printf("\tESC:退出游戏");

gotoxy(SCREEN\_WIDETH + 5, 28);

}

int main()

{

//设置窗口大小

system("mode con cols=110 lines=31");

//设置标题

SetConsoleTitleA("贪吃蛇");

//初始化蛇

while (true) {

snake.key = VK\_RIGHT;//初始化方向

snake.len = 5; //初始化长度

snake.speed = 300;//初始化蛇的移动速度

snake.coord[1].x = SCREEN\_WIDETH / 2; //初始化蛇头的坐标

snake.coord[1].y = SCREEN\_HEIGHT / 2;

snake.coord[2].x = SCREEN\_WIDETH / 2 - 2;//初始化蛇头的坐标

snake.coord[2].y = SCREEN\_HEIGHT / 2;

snake.coord[3].x = SCREEN\_WIDETH / 2 - 4;//初始化蛇头的坐标

snake.coord[3].y = SCREEN\_HEIGHT / 2;

//初始化食物状态

food.flag = 1;//1表示吃到食物 0表示没有吃到食物

//初始化蛇的状态

snake.flag = 1;//1活 0死

init\_sence();//初始化游戏界面

while (1)

{

draw\_snake();//画蛇

Sleep(snake.speed);//蛇的移动速度

move\_snake();//移动蛇

if (food.flag)

creatfood();//产生食物

eatfood();//判断是否吃到食物

SnakeState();//判断蛇是否死亡

if (!snake.flag)break;

}

system("cls");

gotoxy(SCREEN\_WIDETH / 2, SCREEN\_HEIGHT / 2 - 4);

printf("\t游戏结束!!!");

gotoxy(SCREEN\_WIDETH / 2 - 6, SCREEN\_HEIGHT / 2 + 2);

printf("你的得分是：\t\t\t%d ", snake.len - 1);

gotoxy(SCREEN\_WIDETH / 2 - 6, SCREEN\_HEIGHT / 2 + 4);

printf("重新开始：\t\t\tCTRL ");

gotoxy(SCREEN\_WIDETH / 2 - 6, SCREEN\_HEIGHT / 2 + 6);

printf("退出游戏:\t\t\tESC");

while (1)

{

if (GetAsyncKeyState(VK\_CONTROL))

{

system("cls");

break;

}

else if (GetAsyncKeyState(VK\_ESCAPE))

return 0;

}

}

}

//蛇的移动

void move\_snake()

{

//判断是否有按键操作

if (GetAsyncKeyState(up))

{

if (snake.key != down)snake.key = up;

}

else if (GetAsyncKeyState(down))

{

if (snake.key != up)snake.key = down;

}

else if (GetAsyncKeyState(right))

{

if (snake.key != left)snake.key = right;

}

else if (GetAsyncKeyState(left))

{

if (snake.key != right)snake.key = left;

}

else if (GetAsyncKeyState(VK\_F1))

{

if (snake.speed >= 100)snake.speed -= 50;

}

else if (GetAsyncKeyState(VK\_F2))

{

if (snake.speed <= 3000)snake.speed += 100;

}

else if (GetAsyncKeyState(VK\_SPACE)) {

while (1) {

if (GetAsyncKeyState(VK\_CONTROL))

break;

}

}

else if (GetAsyncKeyState(VK\_ESCAPE))

return ;

//根据检测到的方向改变蛇头的位置

switch (snake.key)

{

case right:snake.coord[1].x += 2; break;

case left:snake.coord[1].x -= 2; break;

case up:snake.coord[1].y -= 1; break;

case down:snake.coord[1].y += 1; break;

}

}

//画出蛇

void draw\_snake()

{

//画出蛇头

gotoxy(snake.coord[1].x, snake.coord[1].y);

printf("□");

//画出蛇身，直接一个for循环实现

for (int i = 2; i <= snake.len; i++)

{

gotoxy(snake.coord[i].x, snake.coord[i].y);

printf("□");

}

//擦掉尾巴

gotoxy(snake.coord[snake.len].x, snake.coord[snake.len].y);

printf(" ");

//遍历每一节蛇

for (int i = snake.len; i > 1; i--)

{

snake.coord[i].x = snake.coord[i - 1].x;

snake.coord[i].y = snake.coord[i - 1].y;

}

gotoxy(0, 0);

printf("■");

gotoxy(85, 25);

printf("得分:%d ", snake.len - 1);

}

//产生食物

void creatfood()

{

//随机种子生成

srand((unsigned)time(NULL));

if (food.flag)

while (1)

{

food.x = rand() % 80;

food.y = rand() % 30;

if (food.x % 2 == 0 && food.x >= 2 && food.x <= 78 && food.y > 1 && food.y < 30)

{

int flag = 0;

//判断产生的食物可不可能在蛇的身体上

for (int i = 1; i <= snake.len; i++)

{

if (snake.coord[i].x == food.x&&snake.coord[i].y == food.y)

{

flag = 1;

break;

}

}

if (flag)continue;

//绘制食物

else

{

gotoxy(food.x, food.y);

printf("⊙");

food.flag = 0;

break;

}

}

}

food.flag = 0;

}

//判断蛇是否吃到食物

void eatfood()

{

//只需要判断蛇头是否与食物重合

if (food.x == snake.coord[1].x&&food.y == snake.coord[1].y)

{

snake.len += 1;

food.flag = 1;

}

}

//判断蛇是否死掉

void SnakeState()

{

if (snake.coord[1].x < 2 || snake.coord[1].x>78 || snake.coord[1].y < 1 || snake.coord[1].y>29)

snake.flag = 0;

for (int i = 2; i <= snake.len; i++)

{

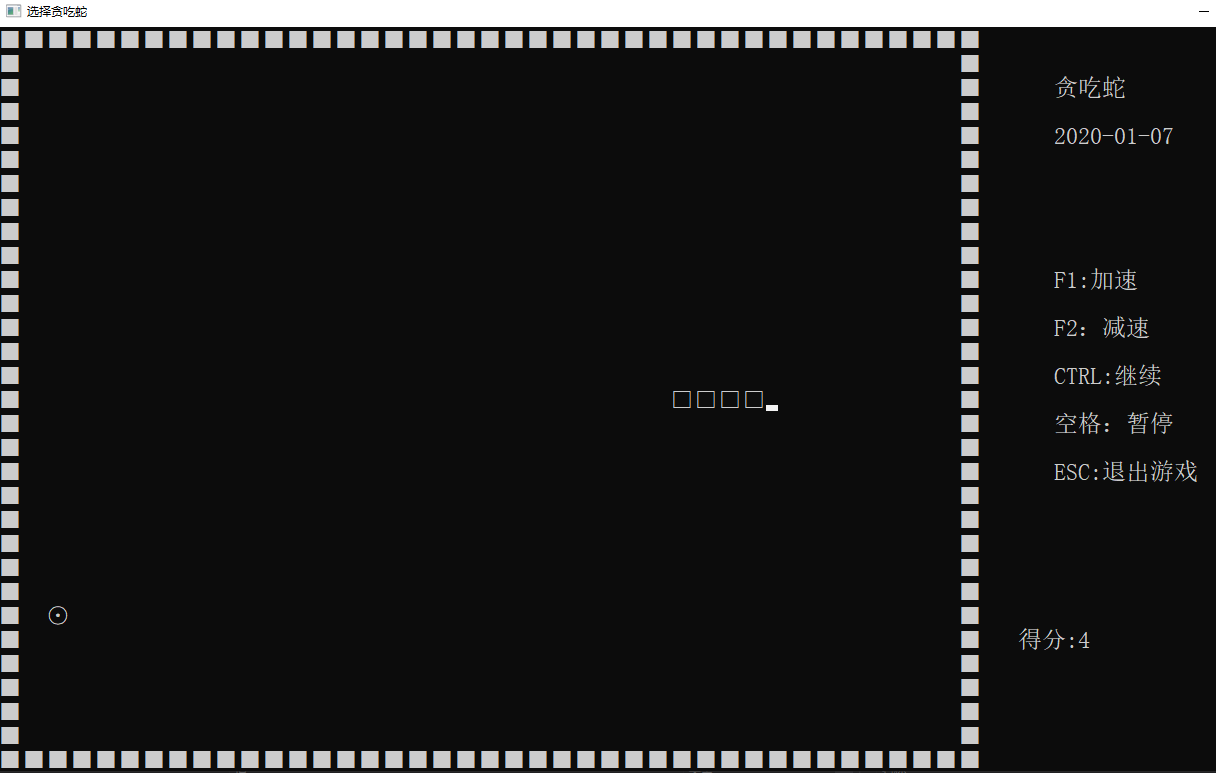
if (snake.coord[1].x == snake.coord[i].x&&snake.coord[1].y == snake.coord[i].y)

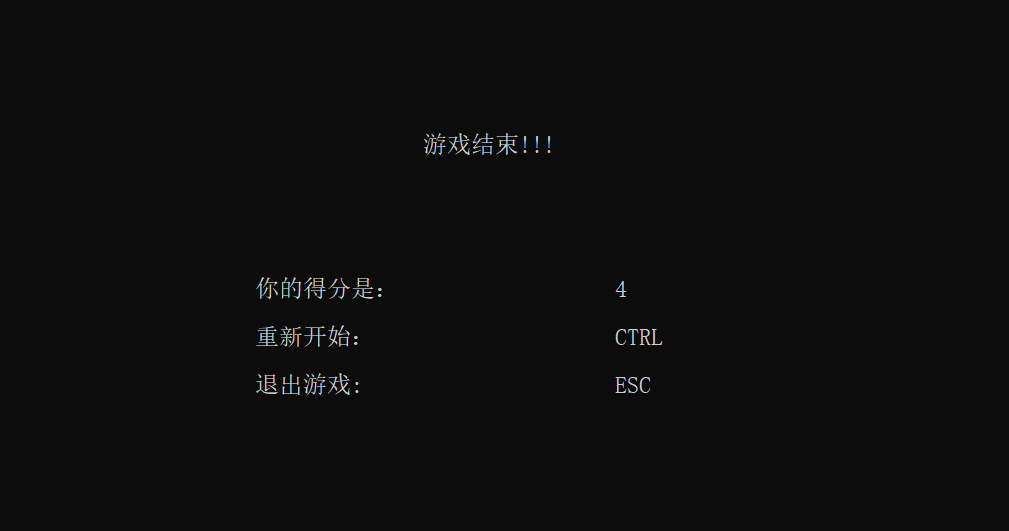
snake.flag = 0;

}

}

**实验结果：**





1. **结论**

通过对贪吃蛇游戏的实现，对C语言有了更深一步的了解。从一开始的小白，自己一点点摸索。从这个实验中，学习到了许多东西。对于贪吃蛇，知道了它的实现热应力，学会了对程序的模块设计、数据结构设计的简单分析。并通过源码分析讲述了贪吃蛇的具体实现。通过本次实现，掌握了如下知识点：键盘操作、结构体定义、函数嵌套和定义等等。总的来说，在本次实验过程中，遇到了许多问题，在编程过程中，遇到了许多bug，从中可以发现，编程是一个需要积累的过程，需要我们静下心来好好学习研究。