C实现贪吃蛇

#include<stdio.h>

#include<stdlib.h>

#include<Windows.h>

#include<time.h>

#include<conio.h>

constexpr auto maphigh = 28, mapwide = 84;

struct virus

{

int x;

int y;

}virus;

unsigned short snakesize = 50, speed = 300, len = 4;

struct snake

{

int x[50];

int y[50];

int len;//蛇的长度

int speed;//蛇的速度

}snake;

int key = 'w';//初始化方向

void gotoxy(int x, int y);

void drawmap();

void keydown();

void creatvirus();

int snakestatus();

void startgame();

int menuselect();

void goodbye();

void introduce();

void edition();

int setup();

void respect();

int setup1();

int modifydiffculty();

void diffculty();

int customize();

void customize1();

int main()

{

for (;;)

{

system("cls");//清屏

switch (menuselect())

{

case 1://开始游戏

startgame();

break;

case 2://介绍

introduce();

break;

case 3://版本

edition();

break;

case 4://设置

setup1();

break;

case 5:

goodbye();

return 0;

default:

break;

}

}

}

void drawmap()//1.画地图

{

//⊙:病毒 █：蛇身(占用两个字符）

srand((unsigned int)time(NULL));//随机病毒出现的位置

int i, k;

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

{

gotoxy(0, i);

printf("█");//打印左边框

gotoxy(mapwide, i);

printf("█");//打印右边框

}

for (i = 0; i <= mapwide; i += 2)//+=2因为 █占两个字节

{

gotoxy(i, 0);

printf("█");//打印下边框

gotoxy(i, maphigh);

printf("█");//打印上边框

}

//画蛇

snake.len = len;

snake.speed = speed;

//初始化蛇的位置

snake.x[0] = mapwide / 2;

snake.y[0] = maphigh / 2;//[0]为蛇头的位置

//画蛇头

gotoxy(snake.x[0], snake.y[0]);

printf("█");

//画蛇身

for (k = 1; k < snake.len; k++)

{

snake.x[k] = snake.x[k - 1] + 2;

snake.y[k] = snake.y[k - 1];

gotoxy(snake.x[k], snake.y[k]);

printf("█");

}

while (1)

{

virus.x = rand() % (mapwide - 4) + 2;//+2,+1这个与█所占的字符有关，长占两个字符宽1个

virus.y = rand() % (maphigh - 2) + 1;//画个图，显而易见

if (virus.x % 2 == 0)

break;

}

gotoxy(virus.x, virus.y);

printf("⊙");

gotoxy(mapwide + 4, maphigh);

printf("⊙:%d", snakesize - snake.len);

}

void creatvirus()//2.产生病毒

{

if (snake.x[0] == virus.x && snake.y[0] == virus.y)

{

//printf("\a");//声音

snake.len++;

srand((unsigned)time(NULL));

while (1)

{

int flag = 1;

virus.x = rand() % (mapwide - 4) + 2;//+2,+1这个与█所占的字符有关，长占两个字符宽1个

virus.y = rand() % (maphigh - 2) + 1;//画个图，显而易见

//产生的病毒不能在蛇的身上

for (int k = 0; k < snake.len; k++)

{

if (snake.x[k] == virus.x && snake.y[k] == virus.y)

{

flag = 0;//virus不合适的标志

break;

}

}

if (flag == 1 && virus.x % 2 == 0)//病毒位置合法且x坐标为偶数

{//x左右是以两个字节为单位运动的，若virus的x坐标为奇数，就不存在snake.x[k] == virus.x

break;

}

}

}

gotoxy(virus.x, virus.y);

printf("⊙");

gotoxy(mapwide + 8, 0);//将光标移走

printf("W");

gotoxy(mapwide + 6, 1);

printf("A S D \t进行控制");

gotoxy(mapwide + 4, maphigh);

printf("⊙:%d", snakesize - snake.len);

}

void keydown()//3.按键

{

int i, temp;

if (\_kbhit()) //kbhit函数检查当前是否有键盘输入，若有则返回一个非0值，否则返回0

{

fflush(stdin);// 清空输入缓冲区，通常是为了确保不影响后面的数据读取

temp = \_getch();//getch:从控制台读取一个字符，但不显示在屏幕上

if ((temp == 'a' || temp == 'A') && (key != 'D' && key != 'd'))//解决了按反方向键蛇自杀的问题

{

key = temp;

}

if ((temp == 'w' || temp == 'W') && (key != 's' && key != 'S'))

{

key = temp;

}

if ((temp == 's' || temp == 'S') && (key != 'W' && key != 'w'))

{

key = temp;

}

if ((temp == 'D' || temp == 'd') && (key != 'a' && key != 'A'))

{

key = temp;

}

}

//擦除最后一节

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

printf(" ");

for (i = snake.len - 1; i > 0; i--)

{

snake.x[i] = snake.x[i - 1];//以前1的位置现在变成了2，0变成了1

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

}

switch (key)//最后一节已经擦出，前移后0,1位置重合,左上角坐标为0，0

{

case'w':

case'W':

snake.y[0]--;

break;

case's':

case'S':

snake.y[0]++;

break;

case'a':

case'A':

snake.x[0] -= 2;

break;

case'd':

case'D':

snake.x[0] += 2;

break;

}

gotoxy(snake.x[0], snake.y[0]);

printf("█");

gotoxy(mapwide + 2, 0);//将光标移走

}

int snakestatus()//4.蛇的状态

{

if ((snake.x[0] == 0 || snake.x[0] == mapwide) || (snake.y[0] == 0 || snake.y[0] == maphigh))

return 0;

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

{

if (snake.x[0] == snake.x[k] && snake.y[0] == snake.y[k])

return 0;

}

return 1;

}

void gotoxy(int x, int y)

{//1.找到控制台这个窗口

HANDLE handle = GetStdHandle(STD\_OUTPUT\_HANDLE);

/\*HANDLE为句柄 ↑得到（输出窗口的）句柄。

Windows是一个以虚拟内存为基础的操作系统，很多时候，

进程的代码和数据并不全部装入内存，进程的某一段装入内存后，

还可能被换出到外存，当再次需要时，再装入内存。两次装入的地址绝大多数情况下是不一样的。

也就是说，同一对象在内存中的地址会变化。那么，程序怎么才能准确地访问到对象呢？为了解决这个问题，Windows引入了句柄。

数值上，是一个32位无符号整型值（32位系统下）；逻辑上，相当于指针的指针；形象理解上，是Windows中各个对象的一个唯一的、固定不变的ID；

作用上，Windows使用句柄来标识诸如窗口、位图、画笔等对象，并通过句柄找到这些对象。\*/

//2.设置光标

COORD coord;

/\*COORD 为Windows.h中自带函数原型大体为struct \_coord{short x;short y;}coord;\*/

coord.X = x;

coord.Y = y;

//4.同步到控制台SetConsoleCursorPosition

SetConsoleCursorPosition(handle, coord);//定位到handle这个窗口，把光标打在coord坐标

}

void startgame()

{

system("cls");

drawmap();

while (1)

{

creatvirus();

keydown();

Sleep(snake.speed);//void sleep(int seconds)自带函数参数 seconds 为要暂停的毫秒数。

if (!snakestatus())//判断死亡时snakestaus为0,

{

gotoxy(mapwide / 2, maphigh / 2);

printf("Game Over");

getchar();

getchar();

break;

}

if (snake.len == snakesize)

{

gotoxy(mapwide / 3, maphigh / 2);

printf("恭喜您消灭了全部病毒，胜利(＾－＾)V");

getchar();

respect();

break;

}

}

}

int menuselect()

{

char number;

int a;

printf("\n\n\t\t\t\t\t\t1.开始游戏\n");

printf("\n\t\t\t\t\t\t2.游戏介绍\n");

printf("\n\t\t\t\t\t\t3.游戏版本\n");

printf("\n\t\t\t\t\t\t4.设置\n");

printf("\n\t\t\t\t\t\t5.退出游戏\n");

printf("\n\t\t\t\t\t\t请选择（数字）");

while (1)

{

number = getchar();

a = (int)number - 48;

if (number <= '5' && number >= '1')

return a;

printf("\n\t\t\t\t\t\t\t ");

}

}

void goodbye()

{

system("cls");

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_INTENSITY | FOREGROUND\_RED);//设置红色

gotoxy(0, 12);

printf("\t\t\t\t\t\t谢谢使用！再见！\n");

getchar();

getchar();

}

void respect()

{

system("cls");

gotoxy(0, 4);

printf("\t\t\*在此向全国的医护人员表示敬意\*\n\n");

Sleep(1000);

printf("\t\t\*感谢他们的默默付出\*\n\n");

Sleep(1000);

printf("\t\t\*感谢他们对抗击疫情做出的贡献\*\n\n");

Sleep(1000);

printf("\t\t\*此致\*\n\n");

printf("\t\t\* 敬礼 \*\n\n");

Sleep(2000);

getchar();

}

void introduce()

{

system("cls");

gotoxy(0, 4);

printf("\t\t\t\t 游戏规则 \n\n");

printf("\t\t\t\t2020年新冠病毒肆虐，威胁着人类\n\n");

printf("\t\t\t\t玩家将控制蛇消灭随机出现的病毒⊙\n\n");

printf("\t\t\t\t消灭所有病毒即可获得胜利。\n\n");

printf("\n\n\n\n\n");

printf("\t\t\t\t\tEnter返回主菜单");

getchar();

getchar();

}

void edition()

{

system("cls");

gotoxy(0, 4);

printf("\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf("\t\t\t\t\* 欢 迎 \*\n\n");

printf("\t\t\t\t\* 版本号： 1.2 \*\n\n");

printf("\t\t\t\t\* 更新：1.修复了反方向自杀的问题 \*\n\n");

printf("\t\t\t\t\* 2.修复了蛇吃墙的问题 \*\n\n");

printf("\t\t\t\t\* 3.新增了菜单与设置功能 \*\n\n");

printf("\t\t\t\t\* 4.修改了部分整形变量节省空间 \*\n\n");

printf("\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf("\t\t\t\t\tEnter返回主菜单");

getchar();

getchar();

}

int setup()

{

//system("color 6f"); //第一个为背景色，第二个为字体颜色

/\*0 = 黑色 8 = 灰色

1 = 蓝色 9 = 淡蓝色

2 = 绿色 A = 淡绿色

3 = 湖蓝色 B = 淡浅绿色

4 = 红色 C = 淡红色

5 = 紫色 D = 淡紫色

6 = 黄色 E = 淡黄色

7 = 白色 F = 亮白色\*/

char s;

int a;

system("cls");

printf("\n\n\t\t\t\t\t\t1.修改难度\n");

printf("\n\t\t\t\t\t\t2.自义定设置\n");

printf("\n\t\t\t\t\t\t3.颜色设置\n");

printf("\n\t\t\t\t\t\t4.音效设置\n");

printf("\n\t\t\t\t\t\t5.返回\n");

printf("\n\t\t\t\t\t\t请选择（数字）");

while (1)

{

s = getchar();

a = (int)s - 48;

if (s <= '5' && s >= '1')

return a;

printf("\n\t\t\t\t\t\t\t ");

}

}

int setup1()

{

switch (setup())

{

case 1:

diffculty();

break;

case 2:

customize1();

break;

default:

break;

}

return 0;

}

int modifydiffculty()

{

char s;

int a;

system("cls");

printf("\n\n\t\t\t\t\t\t1.简单\n");

printf("\n\t\t\t\t\t\t2.普通\n");

printf("\n\t\t\t\t\t\t3.困难\n");

printf("\n\t\t\t\t\t\t4.修罗地狱\n");

printf("\n\t\t\t\t\t\t请选择（数字）");

while (1)

{

s = getchar();

a = (int)s - 48;

if (s <= '4' && s >= '1')

return a;

printf("\n\t\t\t\t\t\t\t ");

}

}

void diffculty()

{

switch (modifydiffculty())

{

case 1:

len = 4;

speed = 500;

snakesize = 10;

system("cls");

gotoxy(8, 8);

printf("\t\t\t\t\t\t 修改成功！\n");

printf("\t\t\t\t\t\t \n");

system("pause");

break;

case 2:

len = 4;

speed = 300;

snakesize = 25;

system("cls");

gotoxy(8, 8);

printf("\t\t\t\t\t\t 修改成功！\n");

printf("\t\t\t\t\t\t \n");

system("pause");

break;

case 3:

len = 4;

speed = 70;

snakesize = 50;

system("cls");

gotoxy(8, 8);

printf("\t\t\t\t\t\t 修改成功！\n");

printf("\t\t\t\t\t\t \n");

system("pause");

break;

case 4:

len = 4;

speed = 25;

snakesize = 70;

system("cls");

gotoxy(8, 8);

printf("\t\t\t\t\t\t 修改成功！\n");

printf("\t\t\t\t\t\t \n");

system("pause");

break;

default:

break;

}

}

int customize()

{

char s;

int a;

system("cls");

printf("\n\n\t\t\t\t\t\t1.自定义速度\n");

printf("\n\t\t\t\t\t\t2.自定义初始长度\n");

printf("\n\t\t\t\t\t\t请选择（数字）");

while (1)

{

s = getchar();

a = (int)s - 48;

if (s <= '2' && s >= '1')

return a;

printf("\n\t\t\t\t\t\t\t ");

}

}

void customize1()

{

int s;

switch (customize())

{

case 1://自定义速度

system("cls");

gotoxy(8, 8);

printf("\t\t\t\t请输入速度(1-999)");

scanf\_s("%d", &s);

speed = (1000 - s);

break;

case 2:

system("cls");

gotoxy(8, 8);

printf("\t\t\t\t请输入初始长度：");

scanf\_s("%d", &s);

len = s;

break;

default:

break;

}

}

————————————————

版权声明：本文为CSDN博主「JiYH」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/qq\_45882032/article/details/108943569