**《程序设计课程实践》设计文档**

# 作业题目：贪吃蛇

**学号：\_\_\_\_\_\_\_\_\_\_\_\_\_\_19151633\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_应宇杰\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

完成时间 2021 年 5 月 11 日

第\_\_7\_\_\_次作业 (写上第几次作业)

题目\_\_\_贪吃蛇\_\_\_\_\_（写上题目号）

提交代码：

#include <cstdio>

#include <windows.h>

#include <conio.h>

#include <ctime>

#include <algorithm>

using namespace std;

const int dx[]={-1,1,0,0};

const int dy[]={0,0,-1,1};

//上下左右

const int H=20,W=40;

struct Node{

int x,y;

inline bool operator == (const Node &lyf){return x==lyf.x&&y==lyf.y;}

}food;

struct Snake{

Node a[1000];

int len;

inline Node &operator [] (const int &x){return a[x];}

}snake;

int now\_dir,dir;

//定位光标

void Goto(short int x,short int y){

swap(x,y);

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

COORD cor=(COORD){x,y};

SetConsoleCursorPosition(hout,cor);

return;

}

void Hide(void){

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

CONSOLE\_CURSOR\_INFO cor\_info={1,0};

SetConsoleCursorInfo(hout,&cor\_info);

return;

}

int Menu(void){

Goto(12,40),printf("贪吃蛇小游戏");

Goto(14,43),printf("1. 开始游戏");

Goto(16,43),printf("2. 帮助");

Goto(18,43),printf("3. 关于");

Goto(20,43),printf("其他任意键退出游戏");

Hide();

char ch=getch();

system("cls");

switch (ch){

case '1': return 1; break;

case '2': return 2; break;

case '3': return 3; break;

default: return 0; break;

}

}

void About(){

Goto(12,30),printf("杭州电子科技大学——程序设计综合实践案例");

Goto(14,43),printf("贪吃蛇小游戏");

Goto(16,43),printf("按任意键返回上级菜单");

Hide();

char ch=getch();

if (ch==(char)-32) ch=getch();

system("cls");

return;

}

void Help(){

Goto(12,40),printf("w或方向键上：上");

Goto(14,40),printf("s或方向键下：下");

Goto(16,40),printf("a或方向键左：左");

Goto(18,40),printf("d或方向键右：右");

Goto(20,40),printf("当蛇撞到自身或撞到墙时游戏结束");

Goto(22,45),printf("按任意键返回上级菜单");

Hide();

char ch=getch();

//方向键有关

if (ch==(char)-32) ch=getch();

system("cls");

return;

}

void CreateFood(void){

bool b=1;

while (b){

b=0;

food=(Node){rand()%(H-2)+1,rand()%(W-2)+1};

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

if (food==snake[i]) b=1;

}

Goto(food.x,food.y),putchar('$');

return;

}

void InitMap(void){

Hide();

snake[0]=(Node){H>>1,W>>1};

Goto(snake[0].x,snake[0].y),putchar('@');

now\_dir=rand()%4,snake.len=3;

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

snake[i].x=snake[i-1].x+dx[now\_dir^1],

snake[i].y=snake[i-1].y+dy[now\_dir^1],

Goto(snake[i].x,snake[i].y),putchar('O');

for (int i=0; i<W; ++i)

Goto(0,i),putchar('-'),

Goto(H-1,i),putchar('-');

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

Goto(i,0),putchar('|'),

Goto(i,W-1),putchar('|');

CreateFood();

return;

}

bool IsDead(void){

//撞到墙

if (snake[0].x==0||snake[0].x==H-1||snake[0].y==0||snake[0].y==W-1) return 1;

//咬到尾巴

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

if (snake[0]==snake[i]) return 1;

return 0;

}

bool MoveSnake(void){

Node tmp=snake[snake.len-1];

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

Goto(snake[1].x,snake[1].y),putchar('O');

if (kbhit()){

char ch=getch(),t;

if (ch==(char)-32)

switch (t=getch()){

case 72: dir=0; break;

case 80: dir=1; break;

case 75: dir=2; break;

case 77: dir=3; break;

}

else {

if (ch=='w'||ch=='W') dir=0;

if (ch=='s'||ch=='S') dir=1;

if (ch=='a'||ch=='A') dir=2;

if (ch=='d'||ch=='D') dir=3;

}

if (dir!=(now\_dir^1)) now\_dir=dir;

}

snake[0].x+=dx[now\_dir],snake[0].y+=dy[now\_dir];

Goto(snake[0].x,snake[0].y),putchar('@');

bool flag=0;

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

flag=1,snake[snake.len++]=tmp;

if (!flag)

Goto(tmp.x,tmp.y),putchar(' ');

else

CreateFood(),Goto(10,50),printf("当前得分：%d",snake.len-3);

if (IsDead()){

system("cls");

Goto(14,45),printf("最终得分：%d",snake.len-3);

Goto(16,45),printf("你输了！");

Goto(18,45),printf("按任意键返回主菜单");

char ch=getch();

if (ch==(char)-32) ch=getch();

system("cls");

return 0;

}

Sleep(100);

return 1;

}

int main(void){

srand(time(0));

while (1){

int tmp=Menu();

switch(tmp){

case 1:{

InitMap();

for (int i=3; i>0; --i)

Goto(10,50),printf("%d后开始游戏",i),Sleep(1000);

Goto(10,50),printf("当前得分：0");

while (MoveSnake());

break;

}

case 2: Help(); break;

case 3: About(); break;

default: return 0;

}

}

return 0;

}





