单按键输入\_打飞机

完整代码：

#include<stdio.h>

#include<stdlib.h> //清屏头文件

#include<conio.h> //不按回车即直接输入数据进程序

#include <stdlib.h> //随机数生成

#include <time.h> //得到当前时间 精确到秒

#include<Windows.h> //睡眠---Sleep(50); //重画屏幕和光标不闪烁

#define m\_DIJI 5 //敌机数量

#define m\_ZHIDAN 10 //子弹上限

#define high 20 //画面高

#define width 20 //画面宽

#define nandu 8 //越小越快

//光标移动到（x，y）位置

void gotoxy(int x, int y) {//重新画屏

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

COORD pos;

pos.X=x;

pos.Y=y;

SetConsoleCursorPosition(handle, pos);

}

void HideCursor() {//解决光标闪烁问题

CONSOLE\_CURSOR\_INFO cursor\_info = { 1,0 };//第二个值为1表示隐藏光标

SetConsoleCursorInfo(GetStdHandle(STD\_OUTPUT\_HANDLE), &cursor\_info);

}

//函数外全局变量

//死亡 //得分

int siwang=0,defen=0;

//定义飞机的位置

int fei\_x, fei\_y;

//画面的大小

struct ZIDAN {//子弹

int zhidan\_x=-1, zhidan\_y=-1;

};

struct DIJI {//敌机

int diji\_x=-1, diji\_y = -1;

};

//全局结构体数组

ZIDAN ZiDan[m\_ZHIDAN]; //子弹

DIJI DiJi[m\_DIJI]; //敌机

//函数

void startup() { //数据的初始化

fei\_x = width / 2; //飞机x坐标

fei\_y = high / 2; //飞机y坐标

HideCursor(); // 引用解决光标闪烁问题

}

void show() { //显示画面

gotoxy(0, 0);

int p = 0, q = 0,d=0;

for (int i = 0; i < high; i++) { //高

for (int j = 0; j < width; j++) { // 宽

if (fei\_x == j && fei\_y == i) { //飞机位置时时更新 //此处使用if 可以让在飞机处不出现子弹

printf("\*");

q++;

}

else {

for (int k = 0; k < m\_ZHIDAN; k++) { //子弹位置时时更新

if (ZiDan[k].zhidan\_x == j && ZiDan[k].zhidan\_y == i) {

printf("|");

p++;

}

}

}//有飞机则无子弹，有子弹则无飞机，飞机为大

for (int k = 0; k < m\_DIJI; k++) { //敌机位置更新

if (DiJi[k].diji\_y == i && DiJi[k].diji\_x == j) { //此处应该有敌机

if (p == 1) { //此处有子弹

DiJi[k].diji\_x = -1; //敌机死亡

defen++; //得分加1

}

else if (q == 1) {//有飞机，

fei\_x = width / 2; //飞机x坐标 //死亡

fei\_y = high / 2; //飞机y坐标 //复活

siwang++; //死亡次数1

//defen = 0; //得分清零

}

else {

printf("@"); //打印敌机

d++;

}

}

}

if (p == 0 && q == 0 && d == 0) { //该坐标无飞机和子弹

printf(" ");

}

else { //该坐标有飞机或子弹或敌机

p = 0;

q = 0;

d = 0;

}

}

printf("\n");

}

printf("得分： %d\n死亡: %d\n", defen,siwang);

}

void ZiDongGengXin() {//自动更新

for (int i = 0; i < m\_ZHIDAN; i++) { //子弹飞行代码 //10颗子弹同时飞行 上限

if (ZiDan[i].zhidan\_y != -1)

ZiDan[i].zhidan\_y--;

}

//更新敌机

int p = 1;

static int speed = 0;

if (speed < nandu)

speed++;

if (speed == nandu) {

srand((unsigned )time(NULL));

for (int j = 0; j < m\_DIJI; j++) { //遍历敌机

if (DiJi[j].diji\_x == -1) { //被击落

DiJi[j].diji\_x = (rand() % width);//重新给敌机一个x值

DiJi[j].diji\_y = -1;

p =p+ 1;

}

else if (DiJi[j].diji\_y != high) { //还未掉下来

DiJi[j].diji\_y++;//y数值加1

}

else {

DiJi[j].diji\_y = -1; //掉下来重新掉

DiJi[j].diji\_x = (rand() % width );//p=0 避免超过最大界面

p =p+ 1;

}

if (j = m\_DIJI - 1) { //no//遍历完成 //避免获得随机数相同

p = 1;

}

}

speed = 0; //时间限制清零

}

}

void ShuRuGengXin() { //输入更新

char anjian; //按键输入

if (\_kbhit()) { //按键后为'1',运行if语句后代码

anjian = \_getch();

if (anjian == 'w') {

fei\_y--;

}

if (anjian == 's') {

fei\_y++;

}

if (anjian == 'a') {

fei\_x--;

}

if (anjian == 'd') {

fei\_x++;

}

if (anjian == 'j') {

for (int i = 0; i < m\_ZHIDAN; i++)//选择未发射弹药

if (ZiDan[i].zhidan\_y == -1) {

ZiDan[i].zhidan\_x = fei\_x;

ZiDan[i].zhidan\_y = fei\_y;

break;

}

}

//if (anjian == ' ') {//暂停功能键

//\_kbhit() = 1;

//}

}

}

int main() {

startup(); //数据的初始化

while (true) //游戏循环实现

{

show(); //显示画面

ZiDongGengXin(); //自动更新

ShuRuGengXin(); //输入更新

}

return 0;

}

单按键输入：

只能一次一按--,多个同按一个生效

#include<conio.h> //不按回车即直接输入数据进程序

void ShuRuGengXin() { //输入更新

char anjian; //按键输入

if (\_kbhit()) { //按键后为'1',运行if语句后代码

anjian = \_getch();

if (anjian == 'w') {

fei\_y--;

}

if (anjian == 's') {

fei\_y++;

}

if (anjian == 'a') {

fei\_x--;

}

}