#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#include<windows.h>

#include<conio.h>

#define SPACE 32

#define LEFT 75

#define RIGHT 77

#define DOWN 80

#define ESC 27

#define Wall 2

#define Box 1

#define Kong 0

#define FACE\_X 29 //界面尺寸

#define FACE\_Y 20 //界面尺寸

void gotoxy(int x,int y); //移动光标

int color(int c); //颜色

void hidden\_cursor(); //隐藏光标

void inter\_face(); //初始化界面

void init\_dia(); //初始化方块信息

void draw\_dia(int base,int space\_c,int x,int y); //覆盖方块

void draw\_kong(int base,int space\_c,int x,int y); //画方块

int pd(int n,int space\_c,int x,int y); //判断是否到底

void start\_game(); //开始游戏

int xc();//消除

void read\_file(); //读写最高记录

void write\_file(); //写最高纪录

int grade=0;//当前分数

int max=0;//最高记录

int nn=0;

struct Face

{

int data[FACE\_X][FACE\_Y+10]; //数值，为1是方块，为0是空格

int color[FACE\_X][FACE\_Y+10]; //对应方块的颜色

}face;

typedef struct Diamonds

{ int space[4][4]; //4\*4矩阵，为1为方块，为0 为空

}Dia;

Dia dia[7][4]; //一维基础7个方块，二维表示旋转次数

int main()

{

system("cls");

system("title 俄罗斯方块");

color(1);

system("mode con cols=60 lines=30"); //窗口宽度高度

hidden\_cursor();

srand(time(NULL));

read\_file();

grade=0;

inter\_face();

init\_dia();

nn=rand()%7;

while(1)

{

start\_game();

}

return 0;

}

/////////////////////////////////////////////////

void start\_game()

{ int n,ch,t=0,x=0,y=FACE\_Y/2-2,i,j;

int space\_c=0;//旋转次数

draw\_kong(nn,space\_c,4,FACE\_Y+3);

n=nn;

nn=rand()%7; //随机生成下一块

color(nn);

draw\_dia(nn,space\_c,4,FACE\_Y+3);

while(1)

{

color(n);

draw\_dia(n,space\_c,x,y);//画出图形

if(t==0)

t=15000;

while(--t)

{ if(kbhit()!=0)//有输入就跳出

break;

}

if(t==0)

{

if(pd(n,space\_c,x+1,y)==1)

{ draw\_kong(n,space\_c,x,y);

x++; //向下降落

}

else

{

for(i=0;i<4;i++)

{

for(j=0;j<4;j++)

{

if(dia[n][space\_c].space[i][j]==1)

{

face.data[x+i][y+j]=Box;

face.color[x+i][y+j]=n;

while(xc());

}

}

}

return;

}

}

else

{

ch=getch();

switch(ch) //移动

{

case LEFT: if(pd(n,space\_c,x,y-1)==1) //判断是否可以移动

{ draw\_kong(n,space\_c,x,y);

y--;

}

break;

case RIGHT: if(pd(n,space\_c,x,y+1)==1)

{ draw\_kong(n,space\_c,x,y);

y++;

}

break;

case DOWN: if(pd(n,space\_c,x+1,y)==1)

{ draw\_kong(n,space\_c,x,y);

x++;

}

break;

case SPACE: if(pd(n,(space\_c+1)%4,x+1,y)==1)

{ draw\_kong(n,space\_c,x,y);

space\_c=(space\_c+1)%4;

}

break;

case ESC : system("cls");

gotoxy(FACE\_X/2,FACE\_Y);

printf("---游戏结束!---\n\n");

gotoxy(FACE\_X/2+2,FACE\_Y);

printf("---按任意键退出!---\n");

getch();

exit(0);

break;

case 'R':

case 'r': main();

exit(0);

case 'S':

case 's': while(1)

{ if(kbhit()!=0)//有输入就跳出

break;

}

break;

}

}

}

}

int xc()

{

int i,j,k,sum;

for(i=FACE\_X-2;i>4;i--)

{

sum=0;

for(j=1;j<FACE\_Y-1;j++)

{

sum+=face.data[i][j];

}

if(sum==0)

break;

if(sum==FACE\_Y-2) //满一行，减掉

{

grade+=100;

color(7);

gotoxy(FACE\_X-4,2\*FACE\_Y+2);

printf("分数：%d",grade);

for(j=1;j<FACE\_Y-1;j++)

{

face.data[i][j]=Kong;

gotoxy(i,2\*j);

printf(" ");

}

for(j=i;j>1;j--)

{ sum=0;

for(k=1;k<FACE\_Y-1;k++)

{

sum+=face.data[j-1][k]+face.data[j][k];

face.data[j][k]=face.data[j-1][k];

if(face.data[j][k]==Kong)

{

gotoxy(j,2\*k);

printf(" ");

}

else

{

gotoxy(j,2\*k);

color(face.color[j][k]);

printf("■");

}

}

if(sum==0)

return 1;

}

}

}

for(i=1;i<FACE\_Y-1;i++)

{

if(face.data[1][i]==Box)

{

char n;

Sleep(2000); //延时

system("cls");

color(7);

gotoxy(FACE\_X/2-2,2\*(FACE\_Y/3));

if(grade>max)

{

printf("恭喜您打破记录，目前最高纪录为：%d",grade);

write\_file();

}

else if(grade==max)

printf("与纪录持平，请突破你的极限！");

else

printf("请继续努力，你与最高记录只差：%d",max-grade);

gotoxy(FACE\_X/2,2\*(FACE\_Y/3));

printf("GAME OVER!\n");

do

{

gotoxy(FACE\_X/2+2,2\*(FACE\_Y/3));

printf("是否重新开始游戏(y/n): ");

scanf("%c",&n);

gotoxy(FACE\_X/2+4,2\*(FACE\_Y/3));

if(n!='n' && n!='N' && n!='y' && n!='Y')

printf("输入错误，请重新输入!");

else

break;

}while(1);

if(n=='n' || n=='N')

{

gotoxy(FACE\_X/2+4,2\*(FACE\_Y/3));

printf("按任意键退出游戏！");

exit(0);

}

else if(n=='y' || n=='Y')

main();

}

}

return 0;

}

//////////////////////////////////////////////

void read\_file() //读取最高记录

{

FILE \*fp;

fp=fopen("俄罗斯方块记录.txt","r+");

if(fp==NULL)

{

fp=fopen("俄罗斯方块记录.txt","w+");

fwrite(&max,sizeof(int),1,fp);

}

fseek(fp,0,0);

fread(&max,sizeof(int),1,fp);

fclose(fp);

}

void write\_file() //保存最高记录

{

FILE \*fp;

fp=fopen("俄罗斯方块记录.txt","r+");

fwrite(&grade,sizeof(int),1,fp);

fclose(fp);

}

///////////////////////////////////////////////

int pd(int n,int space\_c,int x,int y) //判断是否到底

{

int i,j;

for(i=0;i<4;i++)

{

for(j=0;j<4;j++)

{

if(dia[n][space\_c].space[i][j]==0)

continue;

else if(face.data[x+i][y+j]==Wall || face.data[x+i][y+j]==Box)

return 0;

}

}

return 1;

}

void draw\_kong(int base,int space\_c,int x,int y)

{

int i,j;

for(i=0;i<4;i++)

{

for(j=0;j<4;j++)

{

gotoxy(x+i,2\*(y+j));

if(dia[base][space\_c].space[i][j]==1)

printf(" ");

}

}

}

void draw\_dia(int base,int space\_c,int x,int y)

{

int i,j;

for(i=0;i<4;i++)

{

for(j=0;j<4;j++)

{

gotoxy(x+i,2\*(y+j));

if(dia[base][space\_c].space[i][j]==1)

printf("■");

}

}

}

void init\_dia()

{

int i,j,k,z;

int tmp[4][4];

for(i=0;i<3;i++)

dia[0][0].space[1][i]=1;

dia[0][0].space[2][1]=1; //土形

for(i=1;i<4;i++)

dia[1][0].space[i][1]=1;

dia[1][0].space[1][2]=1; //L形--1

for(i=1;i<4;i++)

dia[2][0].space[i][2]=1;

dia[2][0].space[1][1]=1; //L形--2

for(i=0;i<2;i++)

{ dia[3][0].space[1][i]=1;

dia[3][0].space[2][i+1]=1; //Z形--1

dia[4][0].space[1][i+1]=1;

dia[4][0].space[2][i]=1;//Z形--2

dia[5][0].space[1][i+1]=1;

dia[5][0].space[2][i+1]=1;//田字形

}

for(i=0;i<4;i++)

dia[6][0].space[i][2]=1;//1形

//////////////////////////////////基础7个形状

for(i=0;i<7;i++)

{

for(z=0;z<3;z++)

{

for(j=0;j<4;j++)

{

for(k=0;k<4;k++)

{

tmp[j][k]=dia[i][z].space[j][k];

}

}

for(j=0;j<4;j++)

{

for(k=0;k<4;k++)

{

dia[i][z+1].space[j][k]=tmp[4-k-1][j];

}

}

}

}

///////////////////////////////////旋转后的21个形状

}

//////////////////////////////////////

void inter\_face()//界面

{ int i,j;

for(i=0;i<FACE\_X;i++)

{ for(j=0;j<FACE\_Y+10;j++)

{ if(j==0 || j==FACE\_Y-1 || j==FACE\_Y+9)

{ face.data[i][j]=Wall;

gotoxy(i,2\*j);

printf("■");

}

else if(i==FACE\_X-1)

{ face.data[i][j]=Box;

gotoxy(i,2\*j);

printf("■");

}

else

face.data[i][j]=Kong;

}

}

gotoxy(FACE\_X-18,2\*FACE\_Y+2);

printf("左移：←");

gotoxy(FACE\_X-16,2\*FACE\_Y+2);

printf("右移：→");

gotoxy(FACE\_X-14,2\*FACE\_Y+2);

printf("旋转：space");

gotoxy(FACE\_X-12,2\*FACE\_Y+2);

printf("暂停: S");

gotoxy(FACE\_X-10,2\*FACE\_Y+2);

printf("退出: ESC");

gotoxy(FACE\_X-8,2\*FACE\_Y+2);

printf("重新开始:R");

gotoxy(FACE\_X-6,2\*FACE\_Y+2);

printf("最高纪录:%d",max);

gotoxy(FACE\_X-4,2\*FACE\_Y+2);

printf("分数：%d",grade);

}

//////////////////////////////////////////////////

void gotoxy(int x,int y) //移动坐标

{

COORD coord;

coord.X=y;

coord.Y=x;

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

}

//////////////////////////////////////////////////

void hidden\_cursor()//隐藏光标

{

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

CONSOLE\_CURSOR\_INFO cci;

GetConsoleCursorInfo(hOut,&cci);

cci.bVisible=0;//赋1为显示，赋0为隐藏

SetConsoleCursorInfo(hOut,&cci);

}

int color(int c)

{

switch(c)

{

case 0: c=9;break;

case 1:

case 2: c=12;break;

case 3:

case 4: c=14;break;

case 5: c=10;break;

case 6: c=13;break;

default: c=7;break;

}

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), c); //更改文字颜色

return 0;

}