#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <easyx.h>

#include <mmsystem.h>//音乐头

#include <windows.h>

#pragma comment(lib,"winmm.lib")//音乐

#define ROW (16) //定义行列

#define COL (32)

#define ImgSize (40)//图片的尺寸

//定义图片资源

IMAGE imgs[12];

void loadResource()

{

for (int i = 0; i < 12; i++)

{

char imgPath[50] = { 0 };

sprintf\_s(imgPath, "./images/%d.jpg", i);

loadimage(&imgs[i], imgPath, ImgSize, ImgSize);

}

}

bool isfirst = true;//是不是第一次进来

//函数声明

void show(int map[][COL]);

void init(int map[][COL], int Min);

void draw(int map[][COL]);

void mouseMsg(ExMessage\* msg, int map[][COL]);

void boomBlank(int map[][COL], int row, int col);

int judge(int map[][COL], int row, int col, int Min);

int main()

{

////启动开始音乐

//mciSendString("play images/start (2).MP3", 0, 0, 0);

//Sleep(2000);

//mciSendString("close images/start (2).MP3", 0, 0, 0);

long long startTime, endTime; // 用于记录游戏开始和结束时间

startTime = GetTickCount(); // 获取游戏开始时间

// ... (其他代码)

int Min = 0;

int difficulty = MessageBox(NULL, "请选择难度：\n\n（是）简单 (16雷)\n（否）中等 (30雷)\n（取消）困难 (50雷)", "扫雷", MB\_YESNOCANCEL);

if (difficulty == IDYES) {

Min = 16;

}

else if (difficulty == IDNO) {

Min = 30;

}

else if (difficulty == IDCANCEL) {

Min = 50;

}

//创建窗口

initgraph(1280, 640);

////启动开始音乐

mciSendString(" ./images/start.mp3 alias bgm", NULL, 0, NULL);

mciSendString("play bgm", NULL, 0, NULL);

//扫雷地图

int map[ROW][COL] = { 0 };

init(map, Min);

//游戏的主循环

while (true)

{

//处理消息

ExMessage msg;

while (peekmessage(&msg, EX\_MOUSE))

{

switch (msg.message)

{

case WM\_LBUTTONDOWN://鼠标左键和右键点击

case WM\_RBUTTONDOWN:

mouseMsg(&msg, map);//鼠标事件判断函数

int ret = judge(map, msg.y / ImgSize, msg.x / ImgSize, Min);//点击之后完成判断

endTime = GetTickCount(); // 获取游戏结束时间

if (ret == -1)

{

draw(map);//贴雷

int select = MessageBox(GetHWnd(), "菜就多练!", "Low B!", MB\_OKCANCEL);

if (select == IDOK)//再来一把

{

//重新初始化

init(map, Min);

}

else// 退出

{

exit(0);

}

}

else if (ret == 1)

{

double gameTime = (endTime - startTime) / 1000.0; // 将游戏时长转换为秒

char timeStr[100];

sprintf\_s(timeStr, "恭喜你！你用了%.2f 秒扫雷成功！", gameTime);

MessageBox(NULL, timeStr, "游戏结束", MB\_OK);//胜利结算

}

system("cls");

printf("judege:%d\n", ret);

show(map);

break;

}

}

draw(map);

}

//show(map);

getchar();

return 0;

}

void show(int map[][COL])

{

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

{

for (int k = 0; k < COL; k++)

{

printf("%2d ", map[i][k]);

}

printf("\n");

}

}

//初始化数据

void init(int map[][COL], int Min)

{

loadResource();

//设置随机数种子

srand((unsigned)(time(NULL)));

//把map全部初始化为0

memset(map, 0, sizeof(int) \* ROW \* COL);

//随机设置五十个雷 用-1表示

for (int i = 0; i < Min;)

{

//数组有效下标【0，15】【0，31】

int r = rand() % ROW;

int c = rand() % COL;

if (map[r][c] == 0)

{

map[r][c] = -1;

//只有执行了这个代码才成功设置了雷

i++;

}

}

//把以雷为中心的九宫格数据都+1，雷除外

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

{

for (int k = 0; k < COL; k++)

{

//找到雷,并遍历雷所在的九宫格

if (map[i][k] == -1)

{

for (int r = i - 1; r <= i + 1; r++)

{

for (int c = k - 1; c <= k + 1; c++)

{

//对周围的数据加+1

if ((r >= 0 && r < ROW && c >= 0 && c < COL) && map[r][c] != -1)

{

map[r][c]++;

}

}

}

}

}

}

//加密格子

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

{

for (int k = 0; k < COL; k++)

{

map[i][k] += 20;

}

}

}

//绘制

void draw(int map[][COL])

{

int i, k;

//贴图，根据map数组的值来贴对应的图片

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

{

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

{

if (map[i][k] >= 0 && map[i][k] <= 8)//规定范围[0,8]

{

int index = map[i][k]; //0 1 2 3 4 5 6 7 8

putimage(k \* ImgSize, i \* ImgSize, &imgs[index]);//按坐标贴图

}

else if (map[i][k] == -1)

{

putimage(k \* ImgSize, i \* ImgSize, &imgs[9]);//贴雷

}

else if (map[i][k] >= 19 && map[i][k] <= 28)

{

putimage(k \* ImgSize, i \* ImgSize, &imgs[10]);//加密格子

}

else if (map[i][k] >= 39)

{

putimage(k \* ImgSize, i \* ImgSize, &imgs[11]);//标记格子

}

}

}

}

//鼠标操作数据

void mouseMsg(ExMessage\* msg, int map[][COL])

{

//先根据鼠标点击的坐标求出对用的数组的下标

int c = msg->x / ImgSize;

int r = msg->y / ImgSize;//获取鼠标点击坐标

printf("%d %d\n", msg->x, msg->y);

//左键打开格子

if (msg->message == WM\_LBUTTONDOWN)

{

//什么时候点击有效,因为二维数组的值不能每一次点击都减20

if (map[r][c] >= 19 && map[r][c] <= 28)

{

//这个函数只能播放wav格式

PlaySound("./images/ting.wav", NULL, SND\_ASYNC | SND\_FILENAME);

map[r][c] -= 20;

boomBlank(map, r, c);//检测一下是不是空白格子，是，炸开空白格子

isfirst = true;

}

}

//右键标记格子

else if (msg->message == WM\_RBUTTONDOWN)

{

//PlaySound("C:\\Users\\lh168\\Desktop\\saolei\\扫雷\\images\\ting.wav", NULL, SND\_ASYNC | SND\_FILENAME);

PlaySound("./images/ting.wav", NULL, SND\_ASYNC | SND\_FILENAME);

//是否能够标记,即格子还没有打开时可以标记

if (map[r][c] >= 19 && map[r][c] <= 28)

{

map[r][c] += 20;

system("cls");

show(map);

}

else if (map[r][c] >= 39)

{

map[r][c] -= 20;//取消标记，同样使用右键操作

}

}

}

//点击空白格子，连环剥开周围的所有空白格子还有数字

void boomBlank(int map[][COL], int row, int col)//row和col是当前点击格子的下标

{

static bool isfirst = true;//是不是第一次进来

int r, c;

//判断row col是不是空白格子

if (map[row][col] == 0)

{

for (r = row - 1; r <= row + 1; r++)

{

for (c = col - 1; c <= col + 1; c++)

{

if ((r >= 0 && r < ROW && c >= 0 && c < COL) && map[r][c] >= 19 && map[r][c] <= 28)//没越界和没有打开

{

// 每一次调用都会播放一下

if (isfirst)

{

//PlaySound("C:\\Users\\lh168\\Desktop\\saolei\\扫雷\\images\\chou.wav", NULL, SND\_ASYNC | SND\_FILENAME);

PlaySound("./images/chou.wav", NULL, SND\_ASYNC | SND\_FILENAME);

isfirst = false;

}

map[r][c] -= 20;

boomBlank(map, r, c);//递归，产生连锁

}

}

}

}

return;

}

//判断游戏的结束条件,输了返回-1，赢了返回1，没结束返回0

int judge(int map[][COL], int row, int col, int Min)

{

int i, k;

int cnt = 0;

//点到了雷，结束

if (map[row][col] == -1)

{

return -1;

}

//点完了格子，结束，游戏胜利，点开了100-10=90个格子

for (i = 0; i < ROW; i++)//遍历二维数组

{

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

{

if (map[i][k] >= 0 && map[i][k] <= 8)

{

++cnt;

}

}//统计打开格子的数量

}

if (ROW \* COL - Min == cnt)

{

return 1;

}

//点完了格子，赢了

return 0;

}