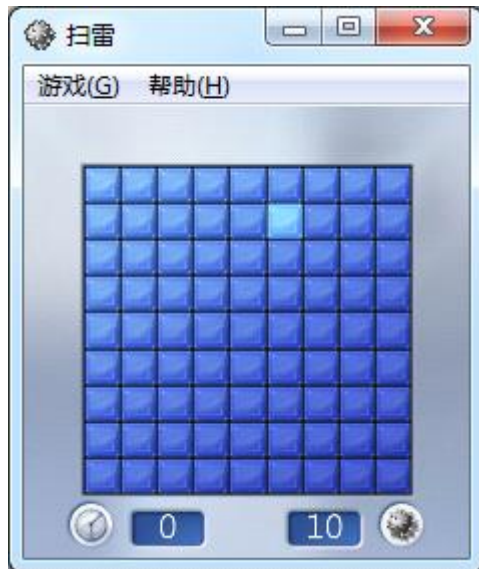


扫雷程序的实现

2014年12月22日 15:27

一直说写个几百行的小项目，于是我写了一个控制台的扫雷，没有想到精简完了代码才200行左右，不过考虑到这是我精简过后的，浓缩才是精华嘛，我就发出来大家一起学习啦，看到程序跑起来能玩，感觉还是蛮有成就感的~哈哈

扫雷应该属于一款大众游戏，从我初中使用计算机开始，它就被集成到了windows系统中，虽然他是这么经典，我还是要介绍一下他的玩法，然后再考虑在控制台中怎么实现它。（扫雷界面 如图1）



- 1 游戏的主界面，是一个一个小方格，在小方格上单击左键，可以翻开小方格看看后面有什么。
- 2 在这些小方格的背后隐藏着雷，如果不幸点中了雷，那么就GameOver了。



- 2 如果点中的不是一个雷，那么就是一块空地，这个时候会出现两种情况：
 - 1) 用鼠标点中的空地周围八个点内有雷，那么就显示雷的个数





2) 用鼠标点中的空地周围没有雷，这个时候就将周围的空地全部显示出来，遇到该显示数字的空地，就将数字显示出来。（仔细观察你会发现，数字会将空地围起来，这是一句废话，但是也值得想一想这是为什么）



3 在小方格上，点击鼠标的右键，可以将一个空地标记为雷，当然这个功能只是为了方便你记忆你之前确定是雷的地方。（还有左右键都点，和点击右键出现？标记，这里就不谈啦）

4 当空地上剩余的格子数和雷的个数一样多，那么这个时候就应该算是胜利啦。

OK~游戏流程说完了，这个时候该谈谈如何实现了。

1 首先需要一张地图，一般情况下我们都可以用一个二维数组表示一个地图，每一个元素代表着扫雷中的一个小方格。相应元素存储0，那么地图上的这个位置就是空地，相应元素存储1，那么就代表这个位置就一颗雷。

2 在控制台上依照二维数组长度和宽度，打印相应的小方块。

3 然后就用鼠标点击那些小方块，对于控制台来讲，在黑框框的区域中是有坐标的，可以使用一些函数捕获到你点击了屏幕的哪一个坐标。

4 对于控制台来说，打印一个字符，有的字符横向占一个位置比如普通的字母数字，有的字符横向占两个位置比如一些图形字符:①②③■◆等等,这点在控制台编程的时候要注意。

5 当点击屏幕的时候,获取到点击的坐标后,去二维数组中查看相应的位置是雷还是空地,从而做相应的处理。

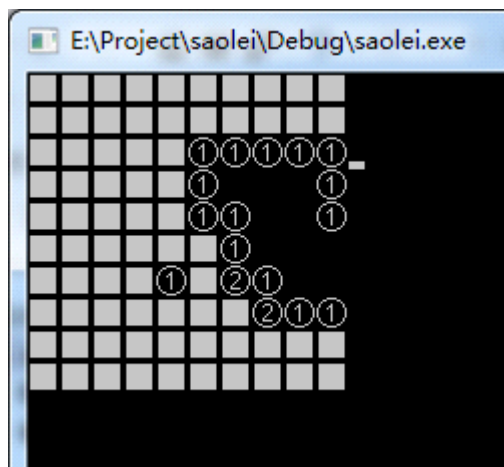
1) 假如点击到了雷，那么就控制游戏结束

2) 假如点击到了空地有两种情况

1) 点击的空地周围有雷，那么就将雷的个数显示出来

2) 假如点击的空地周围没有雷，那么就使用递归的方法去探测周围的点，探测出与其相连的所有周围有雷的点。

这个是我实现的效果：



下面就是代码啦：

```
// saolei.cpp : 定义控制台应用程序的入口点。
//
```

```
#include "stdafx.h"
#include <windows.h>
#include <stdlib.h>
#include <time.h>
#define Boom 10
int a[10][10] = {0};
COORD TempPos[100] = {0};
int nSign = 0;
```

```
/******
```

```
函数名 : WriteWchar
函数作用: 在控制台相应的坐标上显示一串字符
返回值 : void
参数 : int x 横坐标
参数 : int y 纵坐标
参数 : char szString[] 要显示的字符串
说明 :
```

```
*****/
```

```
void WriteWchar(int x, int y, char szString[])
{
    HANDLE hOut= GetStdHandle(STD_OUTPUT_HANDLE);
    COORD pos = {x*2, y};
    SetConsoleCursorPosition(hOut, pos);
    printf("%s", szString);
}
```

```
/******
```

```
函数名 : DrawNumber
函数作用: 在相应的坐标上, 根据传入的数字, 打印相应的数字字符
返回值 : void
参数 : COORD pos 要打印的位置
参数 : int nNumber 要打印的数字
说明 :
```

```
*****/
```

```
void DrawNumber(COORD pos, int nNumber)
{
    switch (nNumber)
    {
    case 1:
        WriteWchar(pos.X, pos.Y, "①");
        break;
    case 2:
```

```

        WriteWchar(pos.X, pos.Y, "②");
        break;
    case 3:
        WriteWchar(pos.X, pos.Y, "③");
        break;
    case 4:
        WriteWchar(pos.X, pos.Y, "④");
        break;
    case 5:
        WriteWchar(pos.X, pos.Y, "⑤");
        break;
    case 6:
        WriteWchar(pos.X, pos.Y, "⑥");
        break;
    case 7:
        WriteWchar(pos.X, pos.Y, "⑦");
        break;
    case 8:
        WriteWchar(pos.X, pos.Y, "⑧");
        break;
    default:
        break;
}
}

/*****
函数名  : GetNumber
函数作用: 获取一个点的四周有几颗雷
返回值  : int
参数    : COORD pos  要探测的点的坐标
说明    :
*****/
int GetNumber(COORD pos)
{
    int nCount = 0;
    for(int i = pos.X-1; i<=pos.X+1; i++)
        for (int j = pos.Y-1; j<=pos.Y+1; j++)
        {
            if (a[j][i] == Boom)
            {
                nCount++;
            }
        }
    return nCount;
}

/*****
函数名  : Drawmap
函数作用: 打印一下地图
返回值  : void
说明    :
*****/
void Drawmap()
{
    for (int i =0; i<10; i++)
    {
        for (int j =0; j<10; j++)
        {
            WriteWchar(j, i, "■");
        }
    }
}

```

```

/*****
函数名  : Init
函数作用: 随机生成10个地雷, 然后存到数组中
返回值  : void
说明    :
*****/
void Init()
{
    srand(time(NULL));
    for (int i =0;i<10;i++)
    {
        int Temp_x = rand()%10;
        int Temp_y = rand()%10;
        //判断这个地方是不是已经生成一个雷了, 如果没有, 赋值为雷
        if (a[Temp_x][Temp_y]!=Boom)
        {
            a[Temp_x][Temp_y] = Boom;
        }
        //如果是雷, 就相当于本次生成没有发生过。。。。
        else
        {
            i--;
        }
    }
    Drawmap();
}

/*****
函数名  : IsClose
函数作用: 判断是不是已经探测过的点, 由于使用的8方向递归的探测, 这样避免重复
返回值  : bool
参数    : COORD posTemp
说明    :
*****/
bool IsClose(COORD posTemp)
{
    for (int i =0;i<nSign;i++)
    {
        if(TempPos[i].X == posTemp.X&&TempPos[i].Y == posTemp.Y)
            return true;
    }
    return false;
}

/*****
函数名  : IsKongdi
函数作用: 判断一个点是空地, 还是雷, 如果是空地, 需要做其他处理
返回值  : void
参数    : COORD pos
说明    :
*****/
bool IsKongdi(COORD pos)
{
    int nNumber = 0;
    //1 如果是雷, 就直接返回一个false说明要挂了
    if (a[pos.Y][pos.X] == Boom)
    {
        return false;
    }
    //2 如果不是雷, 那么就做后续处理

```

```

else
{
    //2.1先判断一下周围有几颗雷
    nNumber = GetNumber(pos);
    if (nNumber!=0) {
        //有几颗雷，就打印这个数字
        DrawNumber(pos, nNumber);
        return true;
    }
    else
    {
        //如果没有雷，那就先画空地出来，然后向周围扩散去探测其他点
        WriteWchar(pos.X, pos.Y, " ");
    }
}

//2.2点到了空地，但是周围没有雷的情况的处理，继续去探测周围8个点
for(int i = -1; i<=1; i++)
    for (int j = -1; j<=1; j++)
    {
        COORD posTemp = {pos.X+i, pos.Y+j};
        //是不是越界了
        if (i==0&&j==0 || posTemp.X== -1 || posTemp.Y== -1 || posTemp.X==10 || posTemp.Y==10)
            continue;
        //这个点是不是已经探测过了
        if (IsClose(posTemp))
            continue;
        //这个点没有探测过，就将其加入到数组中，然后使其在以后的探测中，存入
        TempPos[nSign++] =posTemp;
        IsKongdi(posTemp);
    }
    return true;
}

/*****
函数名  : GetMousePosition
函数作用: 获取鼠标点击的位置，假如没有获取到，就返回(-1, -1)
返回值  : COORD    鼠标点击的坐标
说明    :
*****/
COORD GetMousePosition()
{
    HANDLE hIn = GetStdHandle(STD_INPUT_HANDLE);
    INPUT_RECORD stcInput = {0};
    DWORD buffer;
    COORD pos = {-1, -1};
    ReadConsoleInput(hIn, &stcInput, 1, &buffer);
    if (stcInput.EventType == MOUSE_EVENT)
    {
        MOUSE_EVENT_RECORD stcMouseEnent = stcInput.Event.MouseEvent;

        if (stcMouseEnent.dwButtonState ==FROM_LEFT_1ST_BUTTON_PRESSED )
        {
            pos = stcMouseEnent.dwMousePosition;
            pos.X/=2;
        }
    }
    return pos;
}

int _tmain(int argc, _TCHAR* argv[])
{
    Init();
    COORD pos;

```

```

//开始游戏
while(1)
{
    //获取鼠标点击位置
    pos = GetMousePosition();
    if (pos.X!=-1)
    {
        //如果鼠标点击的位置被探测过了，就开始下一次循环
        if (IsClose(pos))
        {
            continue;
        }
        TempPos[nSign++] =pos;
        bool bIskongdi = IsKongdi(pos);
        //点到雷了，就直接退出游戏了。
        if (false ==bIskongdi)
        {
            system("cls");
            WriteWchar(20,10,"you lose");
            getchar();
            break;
        }
        //检测是不是赢了，赢的条件就是没有被探测的点的个数和雷的个数相等
        if (nSign ==90)
        {
            system("cls");
            WriteWchar(20,10,"you win");
        }
    }
}
return 0;
}

```

