

C语言：扫雷小游戏

经过对C语言一段时间的学习我们可以自己进行一些小项目的编写。

在扫雷游戏中，我们用了模块化设计，将不同的模块放进不同的文件里面分成head.h head.c test.c 三个部分

整个代码最让我感到头疼的是find_mine函数的实现，里面有多个if else语句，并且此函数里面还有其他函数find_mine_count的调用，find_mine_count函数还有一种实现方法是将mine[x][y]周围的八个元素相加减去8*'0'，返回结果。

如下：

```
int find_mine_count(char mine[ROWS][COLS], int x, int y)
{
    return mine[x + 1][y + 1] +
        mine[x][y + 1] +
        mine[x - 1][y + 1] +
        mine[x - 1][y] +
        mine[x - 1][y - 1] +
        mine[x][y - 1] +
        mine[x + 1][y - 1] +
        mine[x + 1][y] - 8 * '0';
}
```

话不多说上代码！

head.c

```
#include "head.h"
void menu(void)
{
    printf("*****\n");
    printf("***** 0. Exit *****\n");
    printf("*****\n");
    printf("***** 1. Play *****\n");
    printf("*****\n");
}

void init_board(char board[ROWS][COLS], int rows, int cols, char set)
{
    for (int i = 0; i <= rows; i++)
    {
        for (int j = 0; j <= cols; j++)
        {
            board[i][j] = set;
        }
    }
}
```

```
    }
}

void display_board(char board[ROWS][COLS], int row, int col)
{
    int i = 0;
    int j = 0;
    printf("-----欢迎来到扫雷游戏-----\n");
    for (i = 0; i <= col; i++)
    {
        printf("%2d ", i);
    }
    printf("\n");
    for (i = 1; i <= row; i++)
    {
        printf("%2d ", i);
        for (j = 1; j <= col; j++)
        {
            printf("%2c ", board[i][j]);
        }
        printf("\n");
    }
}

void set_mine(char board[ROWS][COLS], int row, int col)
{
    int count=mid_count;
    while (count)
    {
        int x = rand() % row + 1;
        int y = rand() % col + 1;

        if(board[x][y]!='0')
        {
            board[x][y] = '1';
            count--;
        }
    }
}

int find_mine_count(char mine[ROWS][COLS], int x, int y)
{
    int count = 0;
    int i = 0;
    for (i = x - 1; i <= x + 1; i++)
    {
        int j = 0;
        for (j = x - 1; j <= x + 1; j++)
        {
            count += (mine[i][j] - '0');
        }
    }
}
```

```
        return count;
    }

void find_mine(char mine[ROWS][COLS],char show[ROWS][COLS],int row, int col)
{
    int win = 0;
    int x = 0;
    int y = 0;
    while (win < row * col - easy_count)
    {
        printf("\n请输入要排除的坐标:");

        scanf("%d %d", &x, &y);

        if (x >= 1 && x <= row && y >= 1 && y <= col)
        {
            if (show[x][y] == '*')
            {
                if (mine[x][y] == '1')
                {
                    printf("\n恭喜你被炸死了! \n");
                    display_board(mine, ROW, COL);
                }
                else
                {
                    int mine_count = find_mine_count(mine, x, y);
                    show[x][y] = mine_count+'0';
                    display_board(show, ROW, COL);
                    win++;
                }
            }
            else
            {
                printf("该坐标已被排查, 请重新输入坐标! \n");
            }
        }
        else
        {
            printf("坐标非法, 重新输入! \n");
        }
    }
    if (win == row * col - easy_count)
    {
        printf("\n很遗憾你把雷全部排查完了, 游戏结束! \n");
    }
}
```

head.h

```
#pragma once
#define _CRT_SECURE_NO_WARNINGS

#include<stdio.h>
#include<stdlib.h>
#include<time.h>

#define ROW 20
#define COL 20

#define ROWS ROW+2
#define COLS COL+2

#define easy_count 10
#define mid_count 25
#define max_count 50
//菜单
void menu(void);

//初始化棋盘
void init_board(char board[ROWS][COLS], int rows, int cols, char set);

//打印棋盘
void display_board(char board[ROWS][COLS], int row, int col);

//布置雷
void set_mine(char board[ROWS][COLS], int row, int col);

//排除雷
void find_mine(char mine[ROWS][COLS], char show[ROWS][COLS], int row, int col);
```

test.c

```
#include "head.h"

void test()
{
    char show[ROWS][COLS] = { 0 };
    char mine[ROWS][COLS] = { 0 };

    init_board(show, ROWS, COLS, '*');
    init_board(mine, ROWS, COLS, '0');

    //display_board(show, ROW, COL);
    //display_board(mine, ROW, COL);

    set_mine(mine, ROW, COL);
    display_board(show, ROW, COL);
    //display_board(mine, ROW, COL);
```

```
    find_mine(mine, show, ROW, COL);

}

int main()
{
    srand((unsigned int)time(NULL));
    int input = 0;
    do
    {
        menu();
        printf("请输入:\n");
        scanf("%d", &input);
        switch (input)
        {
            case 1:
                test();
                continue;
            case 0:
                printf("游戏结束\n");
                break;
            default:
                printf("输入有误, 请重新输入! \n");
                break;
        }
    } while (input);
    return 0;
}
```

这个版本的扫雷游戏只是最简版本，后续还可以添加其他有趣功能：

扫雷游戏的扩展

是否可以选游戏难度

- 简单 9*9 棋盘, 10个雷
- 中等 16*16棋盘, 40个雷
- 困难 30*16棋盘, 99个雷

如果排查位置不是雷, 周围也没有雷, 可以展开周围的一片

是否可以标记雷

是否可以加上排雷的时间显示

