# 扫雷解释文稿

## 【1】常量：

#define COWS 11//棋盘行数

#define ROMS 11//棋盘的列数

#define COUNT\_MINE 9//雷的数量

int cut = COWS \* ROMS - COUNT\_MINE;//判断棋盘上还有几个雷

## 【2】函数的功能：

//布置雷

void setmine(int map[COWS][ROMS], int count\_mine);

//导入图片

void PictureInput();

//在控制台上打印出棋盘

void show(int map[COWS][ROMS]);

//棋盘界面的绘制

void gamedraw();

//获取鼠标信息和其对应的功能

void mouse();

//左键解密

void open(int cow, int rom);

// 游戏的结束

void gameover(int cow, int rom);

## 【3】思路和难点

### 思路：

【1】通过map[cow][rom]数组中的数据来显示不同的盘面

{

map[cow][rom]中>=20未开始盘面

map[cow][rom]中19未开始时含有雷盘面

其他也是类似的

}

【2】通过鼠标的不同情况来实现对map[cow][rom]数组中数据的修改来实现游戏盘面的变化

{

msg.uMsg == WM\_LBUTTONDOWN 鼠标左键落下 map[cow][rom]数组中的数据减20

msg.uMsg == WM\_RBUTTONDOWN 鼠标右键落下 map[cow][rom]数组中的数据加20

}

### 难点：

【1】难点：

在setmine()函数中随机布置雷后要通过两个循环同时改变雷附近的map[cow][rom]中的数据来对应不同的贴图

【2】难点：

在open()函数中要通过递归的方法来将周围没有雷的地方炸开