# 实验报告

## 计算机概论大作业：扫雷游戏设计

**学院：软件学院**

**专业：软件工程**

**班级：软件二班**

**姓名：刘茵**

**学号：1813075**

报告要求

1. 实验题目及要求
2. 实验目的
3. 实验内容
4. 实验步骤及方法
5. 实验结果
6. 实验总结
7. 关于本课程的建议
8. 关于本课程的感想

附录：源代码

1. **实验题目及要求**
   1. 实验题目

编程实现扫雷小游戏

* 1. 实验要求
     1. 基本要求

1. 语言不限。
2. 有扫雷界面，可显示游戏操控步骤、结果等。
3. 有游戏状态的提示，如何输入、游戏成功/失败等。
   * 1. 附加功能
4. 提供难度选择，初级、中级、高级，自定义规模、雷的个数等。
5. 优化界面，提供可视化界面。
6. …
7. **实验目的**

通过C语言编写扫雷程序，娱乐学习结合，提高学习兴趣，学习更多的代码知识。

1. **实验内容**
2. 设置游戏内容

1.1棋盘 定义两个数组

1.2布雷

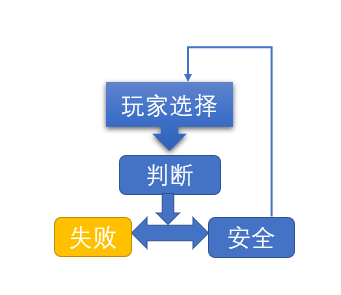
1.3扫雷 排雷

2. 编写游戏代码

3. 代码运行->扫雷游戏->输出游戏结果->选择是否继续游戏

1. **实验步骤及方法**
   1. **写代码之前我在想什么：**

根据作业要求，首先要做的是建立简单的扫雷程序，之后再向内增加功能，初步的要求是：玩家可操作游戏，选择掀开何处位置的棋盘。



并根据游戏失败与否进行相应的提示。

进一步：

玩家可选择扫雷游戏难度->笔者根据扫雷国际标准进行设定。

玩家可再游戏结束是选择是否继续

优化界面，增加游戏可玩性。

* 1. 1. **扫雷之前**：

笔者思考到：

* + - 1. 由八皇后和可以想到定义二维数组作为扫雷游戏的棋盘进行展示，在程序开始之初定义数组。
      2. 棋盘未翻开状态的图形展示char与翻开之后的周围雷数统计int之间的关系时，确定了用两个数组互相转化，便定义一个为玩家查看的棋盘数组***Show***[][]和一个用来储存雷个数的数组***Mine***[][]。
      3. 对反复用到的变量进行全局定义。*Col Row COUNT*等。
      4. 玩家利用输入坐标**（X，Y）**进行选择
    1. **扫雷的开始**
       1. 定义函数start介绍游戏，玩家根据自身情况选择。若玩家离开，则程序关闭，若玩家选择继续 则跳入下个程序 chengdu
       2. 根据游戏难度的不同划分简易 一般 困难三个水平，且定义

行列 雷数

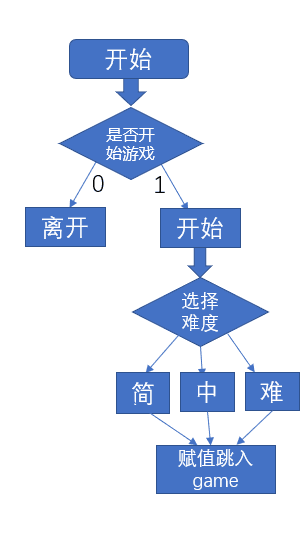
9\*9 10

16\*16 40

16\*30 99

运用switch 进行<1for简单2for一般3for困难>的难度选择 并分别对*Col Row COUNT*进行赋值。

之后跳入game程序方便数组储存元素，打印棋盘等。



* + 1. **扫雷根基1——两数组定义**
       1. 将展示数组***Show***[][]初始化。
       2. 布雷给***Mine***[][]

要求：i.有固定雷数COUNT

ii.生成雷的位置随机

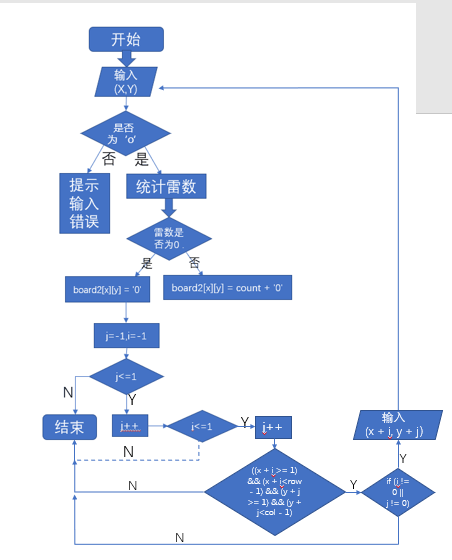
iii.不在同一位置出现两个雷（及生成随机数不相同）

要求一level赋值即可实现，雷数随机笔者学习了srand和rand函数的用法>>根据时间函数产生不同的种子数→产生不同的随机数组。并通过循环得到不重复的（X,Y）坐标赋值给***Mine***。

有雷为’1’,没有雷为’0’方便统计雷数。

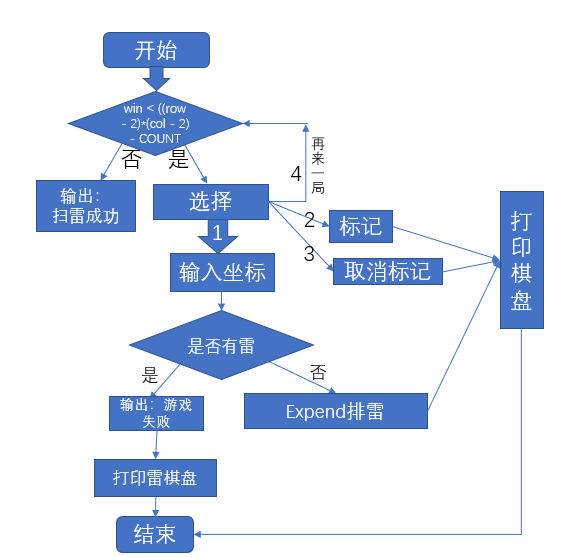
* + 1. **数组打印**
       1. 打印行列坐标方便玩家选择（且根据难度不同当col||row>10时根据字符大小重新布局）
       2. System(“CLS”)函数清屏，让界面更简洁
    2. **扫雷根基2——扫雷展开**

**流程图如下：**

****

* + 1. **开始游戏game()**
       1. 定义一个变量**win**，每当棋盘被翻开而且没有触发到雷时，**win**++,设定game()循环让玩家选择坐标 (X,Y)，当棋盘总格数-WIN<COUNT时，即可判断游戏完成。
       2. **Game**四个选择分别是Sign\notSign\排雷\再来一局
       3. **Sign**和**notSign**只是将棋盘原始符号变换F #F打印棋盘，与**win**无直接联系。
       4. **定义开始时间***starttime***和结束时间***finishtime***：**输出用时。

**流程图**：



**主要错误的改正方法：**

1.sumCOUNT函数统计雷数循环x-1，y-1到x+1，y+1，但当X=0，Y=0时出现越界现象，笔者后将row和col（行列）分别比模式标准大两位，阻止越界，而且此时打印棋盘（X,Y）恰好对应二维数组[X][Y]位，更方便对应。

2.当数组扩增时，随机数的产生应避开周围的四个防越界边框，避免出现雷数<=COUNT的情况。

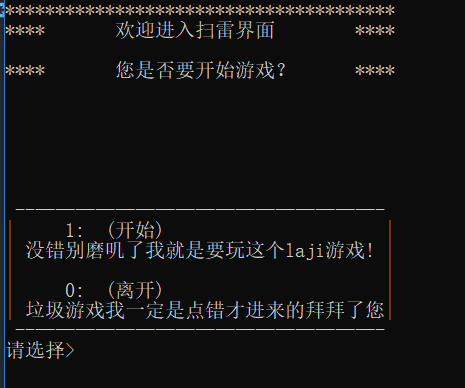
3.定义全局变量

4.排除重复展开的坐标

5．随机数的范围0~n的修正

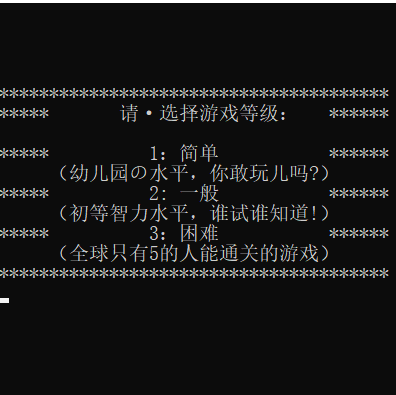
1. 实验结果

（1）



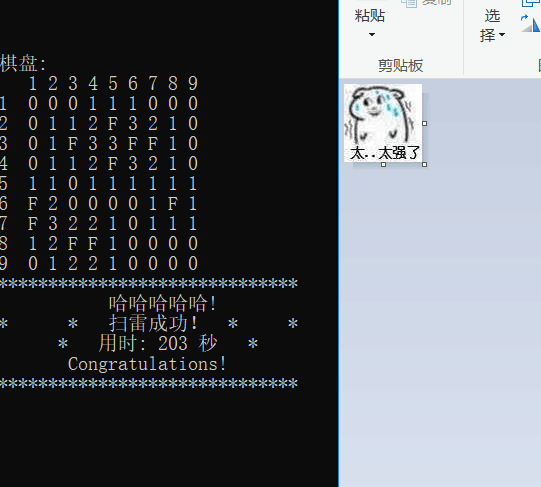
（游戏进入开始界面）

（2）



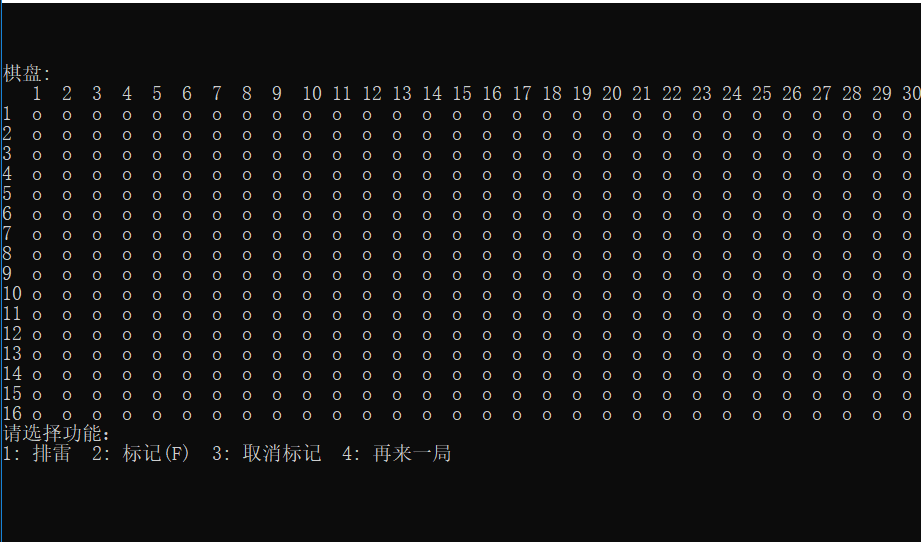
（难度选择页面）

（3）



（简易版的游戏胜利p.s.胜利可弹出图片）

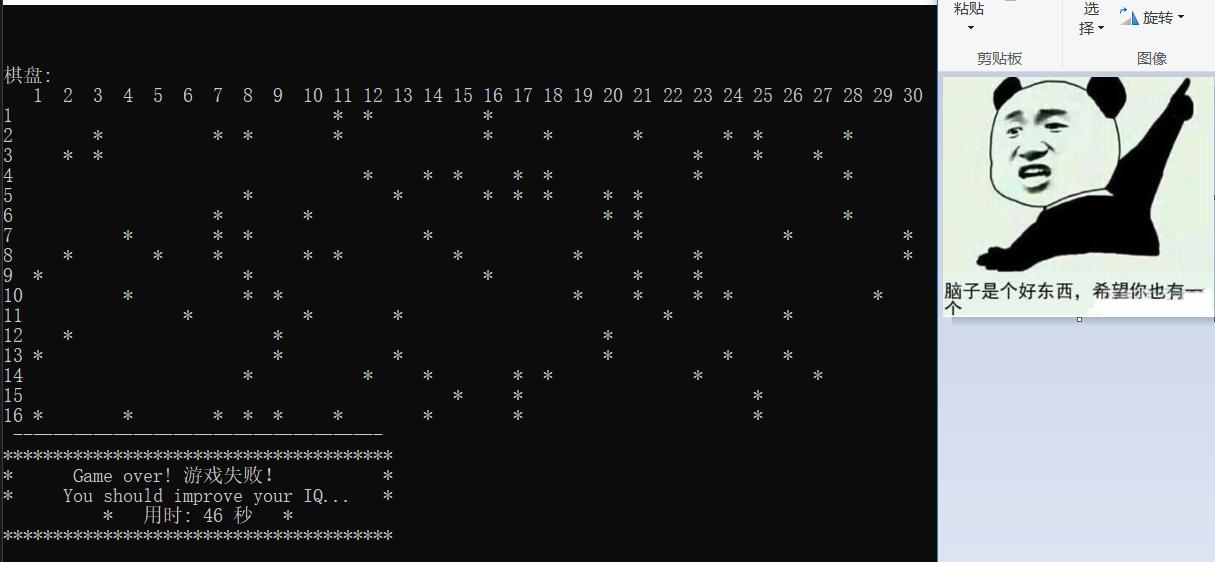
(4)



(困难版游戏界面)

（备注：1234可选）

(5)



（困难版踩雷情况）

1. 实验总结

扫雷游戏的编写，提高了我对C语言理论知识的初步认识和理解，使我掌握了基本的解决编程问题的思路和方法，增加了对程序设计的兴趣。

进行游戏的编写与修正时，我能逐渐熟练的掌握定义函数，调用函数，对递归函数的编写和运用有了更深的认识，也了解了一些以前没有见过的函数，象随机数函数。了解了遇到复杂问题时的解题思路，写出大概结构，然后再具体填充函数。在整个编写过程中，我也做了许多逻辑错误，经多次检查后，不仅锻炼了耐心，还体会到了debug的乐趣与成就感。

通过此次实验，更深刻的认识到我们学习软件工程，不仅应注重理论知识，更应注重实用技能，通过多多实践编写，来提高自己的代码能力。

1. 关于本课程的建议
   * 1. 激发学生的学习兴趣和主动性，为其他专业理论课程的学习做铺垫。增加对最新技术最新成果的简单介绍，增强与学生的课堂互动。
     2. 大作业分小组进行，提升合作意识的同时，可以更多的扩展学生的知识面，集思广益，也可以提高作业完成度，使程序功能开发更充分。
2. 关于本课程的感想
   * 1. 实践比理论更重要
     2. 专业与生活相结合
     3. 专业知识较多，开拓视野和思想界限。
     4. 扎实基础知识，提升编程能力。

附录：源代码

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

/\*用到了time函数，所以要有这个头文件\*/

int nandu; //全局变量

int COUNT; //雷数

int row, col; //行列

char mine[50][50] = { 0 };//存雷的数组（雷盘）

char show[50][50] = { 0 };//展示的数组（雷盘）

double starttime, finishtime;

//行列（难度不同改变row和col的数值）

// 而且行列数比格数多2；

/\*是否开始游戏\*/

void start()

{

void chengdu();

int input = 0;

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\* 欢迎进入扫雷界面 \*\*\*\*\n");

printf("\n");

printf("\*\*\*\* 您是否要开始游戏？ \*\*\*\*\n");

do

{

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf(" --—————————————————-\n");

printf("| 1: (开始) |\n");

printf("| 没错别磨叽了我就是要玩这个<><>游戏! |\n");

printf("| |\n");

printf("| 0: (离开) |\n");

printf("| 垃圾游戏我一定是点错才进来的拜拜了您|");

printf("\n");

printf(" --—————————————————-\n");

printf("请选择>\n");

scanf\_s("%d", &input);

switch (input)

{

case 1:

chengdu(); //游戏程序

break;

case 0:

printf("———— \* 退出游戏 \* ————\n");

printf("———————————————\n");

break;

default:

printf("\* 输入不正确 ，请重新选择！ \*\n");

break;

}

} while (input);

}

/\*选择水平跳入\*/

void chengdu()

{

void game();

system("CLS");

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\* 请·选择游戏等级： \*\*\*\*\*\*\n");

printf("\n");

printf("\*\*\*\*\* 1：简单 \*\*\*\*\*\*\n");

printf(" （幼儿园の水平，你敢玩儿吗?） \n");

printf("\*\*\*\*\* 2: 一般 \*\*\*\*\*\*\n");

printf(" （初等智力水平，谁试谁知道!） \n");

printf("\*\*\*\*\* 3：困难 \*\*\*\*\*\*\n");

printf(" （全球只有5%的人能通关的游戏） \n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

scanf\_s("%d", &nandu); //扫雷难度

switch (nandu)

{

case 1:

row = 11, col = 11;

COUNT = 10;

game();

break;

case 2:

row = 18, col = 18;

COUNT = 40;

game();

break;

case 3:

row = 18, col = 32;

COUNT = 99;

game();

break;

default:

printf("\* 输入不正确 ，请重新选择！ \*\n");

}

}

/\*初始化棋盘并打印\*/

void setchess()//初始化棋盘

{

int i = 0;

int j = 0;

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

{

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

{

show[i][j] = 'o';

}

}

}

/\*打印棋盘\*/

void printboard(char board[50][50]) //行列大于十之后的修改

{

system("CLS");//每次打印雷盘之前清屏一次

int i = 0;

int j = 0;

printf("\n");

printf("\n");

printf("\n");

printf("棋盘:\n");

printf(" ");//为了打印列坐标时对齐

if (row <= 12)

{

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

printf("%d ", i);//打印列坐标1 2 3 4 5 6 7 8 ……

printf("\n");

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

{

printf("%d ", i);//打印行坐标1 2 3 4 5 6 7 8 9 ……

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

{

printf("%c ", board[i][j]);

}

printf("\n");

}

}

else

{

for (i = 1; i < 10; i++)

printf("%d ", i);//打印列坐标1 2 3 4 5 6 7 8 ……

for (i = 10; i < col - 1; i++)

printf("%d ", i);

printf("\n");

for (i = 1; i < 10; i++)

{

printf("%d ", i);//打印行坐标1 2 3 4 5 6 7 8 9 ……

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

{

printf("%c ", board[i][j]);

}

printf("\n");

}

for (i = 10; i < row - 1; i++)

{

printf("%d ", i);//打印行坐标1 2 3 4 5 6 7 8 9 ……

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

{

printf("%c ", board[i][j]);

}

printf("\n");

}

}

}

/\*布雷\*/

void setboard(char board[50][50], int COUNT) //给mine布雷

{

int count = 0;

int x = 0;

int y = 0;

srand((unsigned)time(NULL));//根据时间函数产生不同的种子数→产生不同的随机数组

count = COUNT;//雷数

while (count>0)

{

x = rand() % (row - 2) + 1; //保证行坐标在1到row-2之间 ps: row=行+2

y = rand() % (col - 2) + 1; //保证列坐标在1到col\_2之间

if (board[x][y] == 0)//

{

board[x][y] = '\*';//‘1’代表有雷

count--;//布一次雷，雷数减一

}//如果系统提供的种子没有变化，每次调用rand函数生成的伪随机数序列都是一样的!!

}

}

/\*统计雷数\*/

int sumCOUNT(char board[50][50], int x, int y)//统计雷数

{

int num = 0;

if (board[x - 1][y - 1] == '\*')

num++;

if (board[x - 1][y] == '\*')

num++;

if (board[x - 1][y + 1] == '\*')

num++;

if (board[x][y - 1] == '\*')

num++;

if (board[x][y + 1] == '\*')

num++;

if (board[x + 1][y - 1] == '\*')

num++;

if (board[x + 1][y] == '\*')

num++;

if (board[x + 1][y + 1] == '\*')

num++;

return num;

}

void expend(char board1[50][50], char board2[50][50], int x, int y, int \*num)//扩展式排雷（递归）

{

int i = 0; //1是mine 2是show

int j = 0;

if (board2[x][y] == 'o')//如果该位置字符为'o'，则该位置是未排过的，进行排雷

{

(\*num)++;//排雷次数加一

int count = sumCOUNT(board1, x, y);//统计该位置周围的雷数

if (count != 0)//如果该位置周围的雷数不为0

{

board2[x][y] = count + '0';//显示雷数 切变为字符格式

}

else//如果该位置的雷数为0，则向它周围八个位置扩展排雷

{

board2[x][y] = '0';//该位置的字符显示为‘0’

for (j = -1; j <= 1; j++)

{

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

{

if ((x + i >= 1) && (x + i<row - 1) && (y + j >= 1) && (y + j<col - 1))//确保不越界?

{

if (i != 0 || j != 0)//避免重复排到自己

expend(board1, board2, x + i, y + j, num);

}

}

}

}

}

else

printf("你输入的坐标已经进行了操作，请选择其他坐标进行输入。");

return;

}

/\*雷的标记\*/

void sign(char board[50][50], int x, int y) //用‘F'代表插旗

{

if (board[x][y] == 'o')

{

board[x][y] = 'F';

}

}

void unsign(char board[50][50], int x, int y)//取消标记

{

if (board[x][y] == 'F')

{

board[x][y] = 'o';

}

}

/\*开始游戏\*/

void game()

{

void unsign(char board[50][50], int x, int y);

void sign(char board[50][50], int x, int y);

int sumCOUNT(char board[50][50], int x, int y);

void expend(char board1[50][50], char board2[50][50], int x, int y, int \*num);

void setboard(char board[50][50], int COUNT);

void printboard(char board[50][50]);

void setchess();//初始化棋盘

int x = 0;

int y = 0;

int win = 0;//判定输赢的排雷次数

int select = 0;//决定游戏功能的变量

starttime = clock(); //开始游戏的时间

setchess(); //初始化棋盘

printboard(show);//打印雷盘

setboard(mine, COUNT);//布雷

while (win < ((row - 2)\*(col - 2) - COUNT))//当排雷的次数不少于无雷格数（雷盘格数 减 雷数）时，停止排雷

{

printf("请选择功能：\n1: 排雷 2: 标记(F) 3: 取消标记 4: 再来一局\n");

scanf\_s("%d", &select);

if (select == 1)//1 for 排雷

{

printf("请输入坐标：\n");

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

if (x >= 1 && x <= row - 1 && y >= 1 && y <= col - 1)//检验坐标是否合法

{

if (mine[x][y] == '\*')//如果所选位置有雷

{

if (win == 0)

{

printboard(mine);//排雷失败后打印一下雷的分布

system("mspaint C:\\Users\\master\\Desktop\\ps\\naozi.JPG");

printf(" --—————————————————-\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\* Game over! 游戏失败！ \*\n");

printf("？？？不是我说，你的人品也太差了吧？？？\n");

finishtime = clock();//取结束时间

printf(" \* 用时: %d 秒 \* \n", (int)(finishtime - starttime) / CLOCKS\_PER\_SEC);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n");

printf("Sorry for your stupid……要不你再来一局？\n");

return;

}

else

{

printboard(mine);//排雷失败后打印一下雷的分布

printf(" --—————————————————-\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\* Game over! 游戏失败！ \*\n");

printf("\* You should improve your IQ... \*\n");

finishtime = clock();//取结束时间

printf(" \* 用时: %d 秒 \* \n", (int)(finishtime - starttime) / CLOCKS\_PER\_SEC);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

system("mspaint C:\\Users\\master\\Desktop\\ps\\naozi.JPG");

//插入图片

printf("……要不你再来一局？\n");

return;

}

}

else//如果所选位置没有雷，进行扩展式排雷

{

expend(mine, show, x, y, &win);

}

printboard(show);//打印排雷后的雷盘

}

else

{

printf("错误坐标：\n");

}

}

else if (select == 2)//2 for 标记

{

printf("请输入坐标：\n");

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

fflush(stdin);//清空输入缓冲区，避免多输造成的影响\*/

sign(show, x, y);

printboard(show);

}

else if (select == 3)//3 for 取消标记

{

printf("请输入坐标：\n");

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

fflush(stdin);//清空输入缓冲区，避免多输造成的影响\*/

unsign(show, x, y);

printboard(show);

}

else if (select == 4)//4 for 结束游戏

{

game();

}

}

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf(" 哈哈哈哈哈! \n");

printf("\* \* 扫雷成功！ \* \*\n");

finishtime = clock();

printf(" \* 用时: %d 秒 \* \n", (int)(finishtime - starttime) / CLOCKS\_PER\_SEC);

printf(" Congratulations! \n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

system("mspaint C:\\Users\\master\\Desktop\\ps\\timg.JPG");

}

/\*主函数\*/

int main()

{

start();//开始

system("pause");

return 0;

}