

# Lab4

## 1: 运行结果

(1) 进入程序出现主界面

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
##欢迎来到侯斌洋制作的扫雷游戏! ##
#       s   开始游戏           #
#       h   帮助界面           #
#       q   退出游戏           #
#####
_
```

(2) 在主界面做了恶意输入防范

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
##欢迎来到侯斌洋制作的扫雷游戏! ##
#       s   开始游戏           #
#       h   帮助界面           #
#       q   退出游戏           #
#####
e
输入数据不满足要求, 请重新输入。
2
输入数据不满足要求, 请重新输入。
ren
输入数据不满足要求, 请重新输入。
h_
```

(3) 输入 h 进入帮助界面 (此处为 lab4 要求中的任务 7 自由设计)

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
#####帮助界面#####
#       o x y 打开(x, y)格子   #
#       m x y 标记(x, y)格子   #
#       c x y 取消(x, y)标记   #
# a x y 自动打开(x, y)周围格子 #
#       q   返回主界面         #
#####
```

(4) 之后按 q 返回主界面

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
#####帮助界面#####
#   o x y  打开(x,y)格子   #
#   m x y  标记(x,y)格子   #
#   c x y  取消(x,y)标记   #
# a x y  自动打开(x,y)周围格子 #
#       q   返回主界面     #
#####
q_
```

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
##欢迎来到侯斌洋制作的扫雷游戏! ##
#       s   开始游戏       #
#       h   帮助界面       #
#       q   退出游戏       #
#####
_
```

(5) 然后按 s 开始游戏

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
##欢迎来到侯斌洋制作的扫雷游戏! ##
#       s   开始游戏       #
#       h   帮助界面       #
#       q   退出游戏       #
#####
s
```

(6) 在输入地雷参数时也做了恶意输入防范

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
##欢迎来到侯斌洋制作的扫雷游戏! ##
#       s   开始游戏       #
#       h   帮助界面       #
#       q   退出游戏       #
#####
s
现在请输入雷区的长（小于24）、宽（小于30）和地雷数量
输入格式类似于:9 9 10
a 1 2
输入数据不满足要求，请重新输入。
3 5 b
输入数据不满足要求，请重新输入。
10 10 20_
```

(7) 输入正确数据后打印地雷图（这里显示的地雷和标记数量完成 lab4 要求中的任务 5）

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
地雷数量: 20  标记数量: 0
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
```

(8) 输入 o 3 3 指令打开坐标为 (3,3) 的格子。（注：此处保证第一次打开必不为地雷。）

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
地雷数量: 20  标记数量: 0
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
o 3 3_
```

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
地雷数量: 20  标记数量: 0
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
```

(9) 然后输入 a 5 5 指令自动打开 (5,5) 及其周边无地雷的格子

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
地雷数量: 20  标记数量: 0
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # 2 # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
a 5 5_
```

(10) 这里因为 (5,5) 周围有地雷，故只打开了一个格子。

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
地雷数量: 20  标记数量: 0
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # 2 # # # # # #
# # # # # # # # # #
# # # # # 3 # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
_
```

(11) 然后输入指令 a 0 0 则出现以下结果 (此处完成 lab4 要求中的任务 4)

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
地雷数量: 20  标记数量: 0
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # 2 # # # # # #
# # # # # # # # # #
# # # # # 3 # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
a 0 0
```

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
地雷数量: 20 标记数量: 0
0 1 # # # # # # # #
0 1 # # # # # # # #
0 1 # # # # # # # #
1 1 # 2 # # # # # # #
# # # # # # # # # #
# # # # # 3 # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
```

(12) 接下来输入 o 4 0 (验证失败案例), (4,0) 处恰为地雷, 故判断为失败。

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
地雷数量: 20 标记数量: 0
0 1 # # # # # # # #
0 1 # # # # # # # #
0 1 # # # # # # # #
1 1 # 2 # # # # # # #
# # # # # # # # # #
# # # # # 3 # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
o 4 0
You Lose ~
请按任意键继续. . .
```

注: 在 lab4 中增加了对一次游戏时间的计算, 且无论游戏成功失败与否, 在游戏结束时都会显示本次游戏的时间, 下图为一个例子, 在此说明, 后面部分运行图不进行更换。(此处为 lab4 要求中的任务 7 自由设计)

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
地雷数量: 10 标记数量: 0
0 1 # # # # # # # #
0 1 2 # # # # # # #
0 0 1 2 # # # # # #
0 0 0 1 # # # # # #
0 0 0 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1
0 0 0 0 0 0 2 # #
0 0 0 0 0 0 2 # #
o 0 2
You Lose ~
本次游戏用时 18 s。
Press any key to continue . . .
```

(13) 按任意键后打印完整地雷图，从下图也可以看出 (4, 0) 处确实为地雷。

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
0 1 2 * * 1 1 2 * 1
0 1 * 3 2 1 1 * 2 1
0 1 1 2 1 1 2 2 2 0
1 1 0 2 * 2 1 * 1 0
* 2 1 3 * 2 1 1 1 0
1 2 * 3 2 3 1 1 1 1
1 3 3 4 * 2 * 2 2 *
* 3 * * 3 2 1 2 * 2
1 3 * * 3 1 0 1 1 1
0 1 2 3 * 1 0 0 0 0
请按任意键继续. . .
```

(14) 再按任意键返回主界面。(注：游戏中直接按 q 也可以返回主界面，每次游戏完成后也都会自动返回主界面，然后用户可以选择开始新一轮游戏或在主界面中按 q 退出程序，此处完成 lab4 要求中的任务 6)

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
##欢迎来到侯斌洋制作的扫雷游戏!##
#           s   开始游戏           #
#           h   帮助界面           #
#           q   退出游戏           #
#####
```

(15) 继续进行测试（验证一个特殊的成功案例，同时验证第一个格子必不为雷）

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
##欢迎来到侯斌洋制作的扫雷游戏!##
#           s   开始游戏           #
#           h   帮助界面           #
#           q   退出游戏           #
#####
s
现在请输入雷区的长（小于24）、宽（小于30）和地雷数量
输入格式类似于:9 9 10
8 8 63
```

(16) 由于只有一个雷且第一次不为雷，故直接胜利（此处验证了 lab4 要求中的任务 3）

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
地雷数量: 63  标记数量: 0
#####
#####
#####
#####
#####
#####
#####
#####
o 4 4
You Win ~
请按任意键继续. . .
```

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
*****
*****
*****
*****
*****8*****
*****
*****
*****
请按任意键继续. . .
```

(17) 然后验证普遍情况下的成功案例

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
地雷数量: 5  标记数量: 0
#####
#####
#####
#####
#####
#####
#####
a 5 5
```

(18) 下图展示了标记的操作：输入 m 5 6 标记 (5,6) 位置，并在右上角更新标记数量。同时，输入 c 5 6 可以取消 (5,6) 处的标记（此处完成 lab4 要求中的任务 2）

```

C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
地雷数量: 5  标记数量: 1
0 1 # 1 0 0 0 0
1 2 # 2 1 1 0 0
# # # # # 1 0 0
1 1 1 1 1 1 0 0
1 1 0 0 0 1 1 1
& 1 0 0 0 1 # 1
1 1 0 0 0 1 1 1
0 0 0 0 0 0 0 0
m 5 6

```

```

C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
地雷数量: 5  标记数量: 3
0 1 # 1 0 0 0 0
1 2 2 2 1 1 0 0
# # 1 1 & 1 0 0
1 1 1 1 1 1 0 0
1 1 0 0 0 1 1 1
& 1 0 0 0 1 & 1
1 1 0 0 0 1 1 1
0 0 0 0 0 0 0 0
o 2 0_

```

(19) 下图是一般情况下的成功案例

```

C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
0 1 * 1 0 0 0 0
1 2 2 2 1 1 0 0
1 * 1 1 * 1 0 0
1 1 1 1 1 1 0 0
1 1 0 0 0 1 1 1
* 1 0 0 0 1 * 1
1 1 0 0 0 1 1 1
0 0 0 0 0 0 0 0
请按任意键继续. . .

```



```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
地雷数量: 5  标记数量: 3
0 1 # 1 0 0 0 0
1 2 2 2 1 1 0 0
# # 1 1 & 1 0 0
1 1 1 1 1 1 0 0
1 1 0 0 0 1 1 1
& 1 0 0 0 1 & 1
1 1 0 0 0 1 1 1
0 0 0 0 0 0 0 0
o 2 0
You Win ~
请按任意键继续. . .
```

(20) 此外，在游戏过程中也做了恶意输入防范

(注：在游戏的全部输入过程中都做了恶意输入防范，提示进行正确的输入，此处完成 lab4 要求中的任务 1)

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
地雷数量: 10  标记数量: 0
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
d 3 5
输入数据不满足要求，请重新输入。
2 4 5
输入数据不满足要求，请重新输入。
3 d f
输入数据不满足要求，请重新输入。
o 4 5_
```

```
C:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
地雷数量: 10  标记数量: 0
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # 2 # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
```

以上运行案例完成了 lab4 中的前 6 个任务和 2 个自由设计的功能。

## 2: 代码原理

(1) 首先定义一个类名为 MineMap (地雷图), 其中 private 包含了地雷图的长, 宽, 地雷数量, 以及地雷图二维数组的指针, 辅助标记图二维数组的指针; public 包含了有参构造函数, 无参构造函数, 析构函数, 复制构造函数, 重载赋值运算符, 以及游戏过程中需要对地雷图执行的各种操作函数。该头文件中还包含了程序中用到的非 MineMap 类的其他函数的声明。注: 文件中用 #ifndef 语句防止重复定义 **(在 MineMap.h 文件中)**

(2) 然后就是编写这些函数的代码。对于 MineMap 类的成员函数, 其中无参构造函数默认地雷图长、宽和地雷数量为 9,9,10。有参构造函数需要输入数据。在构造函数中包含了生成随机地雷图的算法, 以保证初始化的每个对象都是随机的。由于 private 中存在二维数组指针, 故也定义了复制构造函数, 重载了赋值运算符以备用, 同时注意在析构函数中用双层 for 循环来 delete 二维数组。然后其他成员函数在注释中都包含了其对应的功能, 具体实现细节也较为简单。AutoOpen 函数的实现用了深度优先搜索, 如果用广度优先搜索的话也是可以的, 不过深度优先代码简洁一些。其实这里如果用哨兵算法的话可以省去很多判断操作, 代码会更加简洁, 但如果用哨兵算法的话我这里就需要改很多东西, 从构造函数开始就需要进行修改; 对于文件中的其他函数, 其实就是把之前 lab3 主程序中的代码分功能搬过来组成一个个能够实现单个功能的函数, 这样可以使主程序看起来更有条理且更加简洁。

**(在 MineMap.cpp 文件中)。**

(3) 最后就是组织主程序了, 组织主程序的主要思路为: ①while 循环控制程序的进行和界面之间的跳转 (界面跳转可能会用到一些标签变量作为 while 循环判断条件); ②由于程序在同一阶段下可能会有不同的输入, 因而在主程序中用 switch 语句来对输入进行选择并根据输入执行不同的操作; ③在不同的功能区合理安排函数的位置以实现内部数据的处理, 同时建立一个较为整洁的用户界面 (可以用 system 函数清屏等操作整理界面); ④根据程

序需要对用户的输入进行引导，同时防范恶意输入；⑤对于每个功能区，存在一些只用函数难以解决的问题，解决这些问题并对细节进行优化，同时做好注释。⑥这里主程序其实可以做成一个类，然后在主程序中只用写几行代码，但我觉得没有必要。再构造一个类的话又要多出两个文件，且主程序中看不到各个界面的内在逻辑关系，还要在其他文件之间跳转。我觉得主程序应该展示程序的主要逻辑。（在 MineSweeper.cpp 文件中）

*以下是一些具体的思路：*

一、游戏总体思路：二维数组 m\_map 存放地雷图，若某处为地雷则设为-1，若不为地雷则用其他数字表示周围的地雷数量。二维数组 m\_mark 存放游戏中的操作，初始全为 0，若打开了某个格子则设为 1；标记某个格子则设为 10；取消标记则设为 0（因为对已经打开的格子进行标记是没有意义的）。根据 m\_map 和 m\_mark 对应位置的关系就可以打印游戏中的地雷图，并判断游戏的成功失败情况。

二、以下是一个检查输入函数的重载，以此例子来说明恶意输入防范的思路（本 lab 中共有 3 个检查输入函数的重载来检查各个界面内的输入）。

```
//检查输入的函数的第三个重载
void InputCheck(bool &CinFail, int &x, int &y, int &length, int &width)
{
    while (x >= length || y >= width || cin.fail() || x < 0 || y < 0)
    {
        cout << "输入坐标不满足要求,请重新输入x和y。" << endl;
        cin.clear();
        while (cin.get() != '\n')
        {
            continue;
        }
        cin >> x >> y;
        CinFail = cin.fail();
    }
}
```

cin 在读取输入后，若输入类型不匹配，则 CinFail 的值就设为 1，这样就防止了不同类型的恶意输入。然后就只需考虑逻辑上的恶意输入，这里利用判断条件使得程序只接受给定范围

的输入，这样也解决了逻辑上的恶意输入，保证了输入的正确性。同时要注意 cin 缓冲区的问题，在下次输入之前要清除错误状态 (cin.clear()) 并清空缓冲区 (while 循环)。

总而言之：①防范类型输入错误；②防范逻辑输入错误；③清空错误状态并重新输入。

三、在本次 lab 中我区分了主界面和游戏界面，并且还做了帮助界面（见上述运行结果），这样在进行一局游戏后可以直接回到主界面进行选择，并且可以查看帮助界面以了解我所定义的游戏操作，另外在对地雷图参数的输入中也对用户进行了引导，这样可以使用户快速上手，不会出现一头雾水的情况。

四、以下代码保证了第一次打开格子不为地雷（此代码被封装到函数 Open (...) 中，此处可暂时忽略前两行的语句进行阅读）

```
else if (a == 'a')
{
    map.AutoOpen(x, y);

    //判断游戏成功失败情况
    // lose==-1失败，win==1成功，lose==win==0继续
    lose = map.JudgeLose();
    win = map.JudgeWin();

    //使第一次打开的格子不为地雷
    while (lose == -1 && first == 0)
    {
        map = MineMap(length, width, MineNum);
        map.AutoOpen(x, y);
        lose = map.JudgeLose();
        win = map.JudgeWin();
        if (lose == 0)
        {
            map.SetTime(); //设置起始时间
            first = 1;
        }
    }
}
if (first == 0)
{
    map.SetTime(); //设置起始时间
}
first = 1;
```

其中 first 是在外部定义的一个 int 型变量，初始值为 0；当外部循环执行第一次时，first 的值为 0，因此可能会进入本函数的循环，此时进行讨论：①第一次打开的格子为雷，则 lose=-1，进入循环，此时重新创建一个对象并将其赋给 map（这里就用到了上面定义过的复制构造函数和重载的赋值运算符），循环一直进行直到新的地图在这个格子上没有地雷，此时将 first 设为 1 结束循环，之后 first 无论经过多少次循环都为 1；因此不会再进入这个循环，防止该循环对正常游戏进行干扰（如果没有 first 标签则游戏永远不会失败）；②第一次打开格子不为雷，则本次不会进入循环，而之后 first 的值又立马被设为 1；故之后的过程中也不会再进入这个循环。

## 五、以下代码实现自动打开周围格子

```
//自动打开周围格子
void MineMap::AutoOpen(int x, int y)
{
    m_mark[x][y] = 1;
    if (m_map[x][y] == 0)
    {
        static int a[3] = {-1, 0, 1};
        static int b[3] = {-1, 0, 1};
        for (int m = 0; m < 3; m++)
        {
            for (int n = 0; n < 3; n++)
            {
                if (x + a[m] >= 0 && x + a[m] < m_length && y + b[n] >= 0 && y + b[n] < m_width && m_map[x + a[m]][y + b[n]] != 10 && m_map[x + a[m]][y + b[n]] != 1)
                {
                    m_mark[x + a[m]][y + b[n]] = 1;
                    MineMap::AutoOpen(x + a[m], y + b[n]);
                }
            }
        }
    }
}
```

用深度优先搜索找出二维数组 m\_map 中为 0 且二维数组 m\_mark 中不为 1（这步判断十分重要，可以防止函数一直递归下去）和 10（若为 10 则已进行标记，故不会自动打开）的格子并打开其周围的格子完成自动打开格子操作。

### 3：总结

这 4 次 lab 练习了类的定义与使用，算是接触 C++ 后写的第一个比较大的程序。刚开始写 MineMap 类的时候还是照着书上的代码案例列出框架，然后具体不会或者不熟悉的地方再去网上找具体的解决方案。由于本次 lab 中 3 个文件在一起编译，在 vscode 中如果是用 C/C++ Runner 插件或者 Code Runner 插件的话需要修改一些配置，不过也可以去网上找别人配置好的 launch.json、task.json 文件作为参考然后进行修改。在学习的过程中老师讲了很多关于封装和重构的问题，在本次 lab 中如果不进行封装只在 MineSweeper.cpp 文件中写全部代码的话要 1000 多行代码，而只是把地雷图封装成一个类然后把其他步骤也整合成一个个函数就可以把主程序缩短为 100 行左右。这样在主程序中看到的只是各个界面而具体操作则隐藏起来，会很有条理。不过我认为主程序也并非越短越好，一些主要逻辑和界面还是有必要保留的。另外写完了代码过一段时间再想办法重构的话确实可以让代码更加整洁，另外注释也应该及时写上、及时修改。