



吉林大学 2019-2020 学年第 2 学期

《数据结构》课程设计

A 题 扫雷游戏算法设计与实现 (本题程序 100 分, 包含 3 小题, 持续 2 周)

扫雷是一款经典游戏, 也是 Windows 操作系统最早引入的一款游戏, 其最为经典的版本是 Windows XP 操作系统自带的扫雷游戏 (见附件 1), 曾风靡一个时代, 是一代人的永恒记忆。



简单来说, 其游戏界面由 n 行 m 列方格组成, 其中 k 个方格后面隐藏着地雷。当用户点击一个方格时:

- (1) 如果该方格后面隐藏着地雷, 则游戏结束, 用户失败。
- (2) 如果该方格不是地雷, 则该方格被打开:
 - 若该方格与地雷相邻 (即该方格上、下、左、右、左上、左下、右上、右下相邻的 8 个方格内有地雷), 则该方格处显示一个数字, 表示其周围 8 个方格中的地雷数。
 - 若该方格未与地雷相邻 (即该方格周围 8 个格子内没有地雷), 则该方格的未被打开的邻居 (即与该方格上、下、左、右、左上、左下、右上、右下相邻的方格)、邻居的邻居、邻居的邻居的邻居……都会被逐级打开, 直到某方格与地雷相邻。这期间每个方格的处理方式同 (2)。
- (3) 当所有没隐藏地雷的方格均被打开 (即所有没打开的方格后都有地雷), 则游戏结束, 用户获胜。

备注: (1) 大家可通过附件 1 中的真实扫雷游戏软件, 熟悉该游戏的玩法。(2) 本题学生不允许使用 STL。

(1) 实现扫雷游戏程序

Time Limit: 4000MS

Memory Limit: 512MB

请编写程序从初始界面开始，对于一系列用户的点击，求出点击之后的游戏界面。未打开的方格用-1表示，即游戏初始时为 n 行 m 列-1。已打开且未与雷相邻的方格用0表示，已打开且与雷相邻的方格用数字 a ($1 \leq a \leq 8$) 表示，即与之相邻的地雷数。

输入格式：

输入第一行是4个正整数 n 、 m 、 k 和 l ，其中 n 、 m 、 k 的含义如前所述。接下来 k 行，每行2个整数 i 和 j ，表示每个雷的坐标，即雷在第 i 行第 j 列的方格里。接下来 l 行，每行2个整数 i 和 j ，表示用户点击信息，即用户点击了第 i 行第 j 列的方格。 m, n 不超过20， k 不超过50， l 不超过200， $0 \leq i < n$ ， $0 \leq j < m$ 。

输出格式：

对于用户的每个点击：（1）如果用户点击的方格是已被打开的方格，则点击无效，忽略该点击。（2）如果点击的方格是地雷，则输出“You lose”，程序退出；（3）如果点击的方格不是地雷，则输出点击后的游戏界面，即 n 行 m 列空格间隔的整数，此时若用户获胜，则再输出“You win”。注：对用户每个有效点击所输出的信息用一个空行间隔。

样例：

输入	输出
5 5 1 1 0 0 4 4	-1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 You win
4 5 1 2 1 2 3 0 1 2	0 1 -1 1 0 0 1 -1 1 0 0 1 1 1 0 0 0 0 0 0 You lose
4 5 1 2 1 2 3 0 0 2	0 1 -1 1 0 0 1 -1 1 0 0 1 1 1 0 0 0 0 0 0 0 1 1 1 0 0 1 -1 1 0 0 1 1 1 0 0 0 0 0 0 You win

提交方式：

学生通过在线评测系统 jlu.openjudge.cn 提交代码，别忘了用自己真名，并选定班级。

评测方法：

该小题满分30分，为全自动评测。本题不采用上学期的ACM竞赛规则，而是采用OI竞

赛规则,该题后台共 6 组测试数据,上学期是必须 6 组全过才算通过;而本次是每组数据 5 分,6 组数据全部通过可得 30 分,只过 5 组则得 25 分,过 4 组得 20 分,以此类推。提交后是 Accepted 则得满分,若是 Wrong Answer,则将页面拉到最下边,能看到自己本次提交的具体得分。可以多次提交,以最后一次提交为准。并不是很看重题目完成的时间。

(2) 智能扫雷算法设计与实现

请继续编写程序,帮助(替代)玩家自动扫雷,用尽可能少的步数最快完成扫雷。请注意,作为玩家,你的程序并不知道地雷的实际位置。本小题无需编写整个程序,只需实现如下函数:

```
void machine(int GamePanel[30][30], int n, int m, int &x, int &y);
```

该函数功能为读入当前游戏界面,并给出决策结果,即在当前游戏界面下,下一步应该点击哪个方格。其参数含义为:数组 GamePanel[][]存储 n 行 m 列整数,表示当前游戏界面,数组中数值的含义同(1)小题。x、y 表示下一步应点击的方格的行号和列号, $0 \leq x < n$, $0 \leq y < m$ 。

也就是说, x 和 y 即本函数的决策结果,表示在当前游戏界面下,应该点击哪个方格。注意在本题中, machine 函数并不知道后台地雷的实际位置,只能根据当前的游戏界面进行决策。本题不允许自行定义全局变量,且 machine 函数只能读 GamePanel 数组,不允许修改该数组,即便修改了该数组,下次调用时修改信息也将丢失。

提交方式:

请同学们通过“超星平台”以“交作业”的方式提交你编写的函数:将你编写的函数存入文本文件中,文件名为“班级-姓名.txt”,如“5 班-张无忌.txt”,以附件形式提交。注意 machine 函数一定严格按照上述接口,函数名和参数不允许有任何改动。若 machine 函数中调用了你自定义的其他函数,则其他函数也须提交,并放在 machine 函数之前,请注意不要提交诸如“#include 头文件”、main 函数等。函数中不允许通过 printf、cout 等输出信息。

评测方法:

该小题满分 60 分,半自动评测。老师编写了评测程序,评测程序采用 20*20 的游戏界面,包含随机放置的 50 个地雷。评测程序不断调用 machine 函数,向其传送当前游戏界面 GamePanel,并获取下一步应点击的方格坐标 x 和 y,并进行点击。直至游戏结束或点击次数超过 400 次。对于每局游戏的首次点击,评测程序先接收 machine 函数的首次点击结果再随机步雷,保证玩家首次点击肯定不会踩雷。

评测程序仅包含 iostream.h、stdio.h、time.h、stdlib.h、math.h 头文件,即你的 machine 函数只允许使用这些头文件内的库函数。

老师收集本班学生代码,将代码放入评测程序中,在本地运行评测程序得出每名同学的分。每局游戏得分计算方法如下:

- 如果扫雷失败或点击次数超过 400 次游戏仍未结束: **得分=总共打开的方格数/10**。
即打开的格子越多得分越多,因为有 400 个方格,50 个地雷,所以最多能打开 350 个方格,也就是说扫雷失败得分不会超过 35 分。

- 如果扫雷成功：**得分=50+bonus**。其中 $bonus = \begin{cases} 10, & 0 < cnt < 80 \\ 9, & 80 \leq cnt < 90 \\ 8, & 90 \leq cnt < 100 \\ 7, & 100 \leq cnt < 120 \\ 5, & 120 \leq cnt < 150 \\ 3, & 150 \leq cnt < 200 \\ 1, & 200 \leq cnt < 250 \\ 0, & cnt \geq 250 \end{cases}$

cnt 为有效点击数，例如在点击 80 次以内完成扫雷，则得 60 分。即用越少的步数完成扫雷者得分越高，扫雷成功至少得 50 分，至多得 60 分。

- 若提交的代码放入评测程序后编译失败或无法正常运行：**得分=0**。
- 为公平起见，评测程序将进行 **1000 次完整游戏**，取**平均分*1.3**作为最终成绩，最终成绩超过 60 分按 60 分记。
- 本小题成绩将向所有学生公示，接受质疑，若有同学觉得某同学平时水平比自己低，但此次分数却比自己高，有抄袭嫌疑，请与老师联系。除机器查重之外，必要时老师将对可疑学生进行**视频答辩**，若答辩未通过则按作弊处理。
- 因**未按题目要求提交程序**导致成绩为 0 的学生，允许有最多 1 次的修改后重交的机会，但得分将减去 10 分。例如重交的程序得 55 分，则最终成绩为 45 分。
- 对于**之前已经提交作业的同学**，无需再次提交，老师会按新规则重新判分，当然如果你想继续完善程序，可以与老师联系，老师可以给你打回重做。

提示：

本题没有标准答案，同学们可以充分发挥想象力给出可能的解法。任何基础、任何层次的同学都有能力给出解答。比如实在想不出好办法的同学，可以随机选择一个方格，即相当于在游戏盘面上随机点击方格，这样虽然很难成功完成扫雷，但至少能打开一定数量的格子，获得一定分数。对于有较好编程基础的同学，可以思考一下，你在实际玩这个游戏的时候，是怎么玩的，采用什么策略，那么就可以把该策略编程实现出来。自己想在本地测试 `machine` 函数的效率怎么办？可以自己编写一个主程序，也可以利用接下来的(3)小题。

(3) 图形界面扫雷游戏的实现

前两小题已经完成了扫雷的核心算法，但均为控制台程序，运行结果不够直观。为此老师向同学们提供扫雷游戏的图形界面框架（附件 2，开发环境 Visual Studio 2017/ VC MFC），大家无需掌握图形界面编程技术，只需按接口要求，将自己编写的(1) (2)小题程序放入框架，即可编译生成图形界面游戏，具体方法如下：

① 将(1)题中的核心功能封装成如下函数：

```
int RefreshGamePanel(int GamePanel[30][30], int x, int y, int mine[30][30], int n, int m, int k)
```

该函数功能是根据当前游戏界面和玩家点击的方格，给出点击之后的游戏界面。各参数含义为：数组 `GamePanel[][]` 存储 n 行 m 列整数，表示当前的游戏界面， x 和 y 表示玩家点击的方格坐标，即第 x 行第 y 列的方格， $0 \leq x < n$ ， $0 \leq y < m$ 。数组 `mine[][]` 存储地雷的信息，若第 i 行第 j 列有雷则 `mine[i][j]=1`，否则 `mine[i][j]=0`， k 为雷的总数。

该函数根据玩家点击的方格，按照游戏规则更新 `GamePanel[][]` 数组。函数返回值表示游戏

是否结束：若玩家点中的方格是雷，则游戏以失败结束，函数返回-1；若玩家点击方格后，所有非雷方格都被打开，则游戏以获胜结束，函数返回 1；若点击方格后游戏未结束，则返回 0。

② 点击 **Cbutton.sln 文件打开整个项目**，将 RefreshGamePanel 函数和(2)小题实现的 machine 函数放入 CButtonView.cpp 文件中，具体位置已在该文件中标出。

③ 若你的函数中还调用了自定义的其他函数，则这些函数也须一并放入 CButtonView.cpp。

也可将该图形界面游戏作为开发(1)(2)时的测试之用。在已打开的方格处点击鼠标右键，可显示雷的真实位置，进而可以测试你的 RefreshGamePanel 和 machine 函数是否正确。老师给出一个简单示例，请见附件 3。



提交方式：

请同学们通过“超星平台”以“交作业”的方式提交自己的 CbuttonView.cpp 文件，注意无需改文件名，直接提交 CbuttonView.cpp 文件。

评测方法：

本小题 10 分。人工评测。老师在下载你的代码并在本地检查并运行，无论自动扫雷效率如何，只要无明显 bug 本小题即可给满分。