

1. 题目

题目要求分步实现内部数组版和伪图形界面版扫雷游戏。

1.1 内部数组版

1. 选择难度并显示内部数组（含行列号标，雷的数目必须要足够）

2. 输入初始位置，保证初始位置坐标为 0，显示初始位置打开后的内部数组

输入的行列号严格保证大小写，点开的区域亮色返显，找初始区域要写成一个函数，可以是递归方式，也可以是非递归方式

3. 内部数组基础版，可以打开合法的输入位置，根据游戏局面判断游戏结果

4. 在 3 的基础上，增加查看游戏时间，标记某个合法位置为雷，取消已标记为雷的某个位置的功能

1.2 伪图形界面版

5. 画出伪图形界面框架并显示内部数据

6. 在伪图形界面上移动鼠标，并判断鼠标位置（如位置不合法，在某行某列）

7. 2 的伪图形界面版

8. 3 的伪图形界面版

9. 4 的伪图形界面版

1.3 退出

结束程序

2. 整体设计思路

简单来说，程序执行分为两步。

第一步：调用菜单，对用户输入进行对应的判断。

第二步：执行用户输入操作。

第一步菜单函数即可实现。

前四个选项与内部数组版实现相关，后四个选项与伪图形界面版本相关。

两个版本在实现思路上非常相近。由于 1/2/3 和 5/6/7/8 选项分别都是在为完整版的 4(内部数组版)/5(伪图形界面版)选项做铺垫，因此下面介绍的思路为完整版的思路。内部数组版与伪图形化界面版的总体思路基本一致。

2.1 初始化部分

选择难度并展示该难度下被遮盖的扫雷地图。

根据用户选择的第一个开始的位置生成扫雷的内部界面

2.2 输入部分

判断用户输入是否满足要求，得到符合要求的输入内容。

2.3 记录部分

记录地图上每个点的状态。状态可分为三种：未被访问且未被标记为雷，已被访问并展示，未被访问但被标记为雷。

2.4 判断输出部分

根据用户的输入判断：如果是点击/查看某位置，查看位置合法时，则按游戏规则展开能看到的界面；如果是查看时间，则显示对应时间；如果是标记或取消标记且位置合法，则进行对应的操作。判断游戏是否结束，并输出胜利/失败。如果用户选择退出则退出游戏。

2.5 返回部分

清屏并根据用户输入返回菜单。

3. 主要功能的实现

篇幅限制，以下涉及函数均省略参数

map数组为int型，记录周围的雷的数量，如果该位置是雷，则该位置为9

visited数组为int型，辅助标记对应的位置的访问状态，如果未被访问且未被标记为雷，则值为0；如果已被访问过，即显示在窗口上，则值为1；如果未被访问且被标记为雷，则值为2

3.1 初始化部分

涉及函数：bool init_model //整个初始化

void count_boom//计算周围炸弹数目

void set_map//放置炸弹

内部数组版：void model_select//选择难度

bool row_col_select//选择行列

int complex_select_base//选择行列\时间\标记\取消标记

伪图形界面版：int graph_row_col_select//选择行列\时间\标记\取消标记

地图的生成：根据难度模式随机生成对应数目的雷，如果雷生成的位置在用户选择的初始位置上或初始位置附近，或者该位置已经生成过雷，则跳过该位置继续生成，直到数量足够。对于未生成雷的位置，计算其周围方向（最多8个方向）的雷的数目存入map数组。

3.2 根据选择的位置显示内部地图

涉及函数：

内部数组版：void DFS_display_unhide

void display_unhide

伪图形界面版：void DFS_draw_unhide_graph

void DFS_draw_unhide_graph

如果该位置是雷，则打印雷，返回；如果该位置不是雷则调用对应的DFS，DFS是一个递归函数，在遍历的每一层，如果该层是雷，则返回；如果该层map对应的值不是0也不是雷，则显示对应的数字并返回；如果map对应的值是0，则显示该位置并继续DFS搜索其周围（最多8个方向）未被遍历过且未超过地图范围的点。

3.3 游戏部分

涉及函数：void display_time//显示时间

```
int judge_game//判断并返回游戏状态  
int judge_row//返回当前难度模式下的行数  
int judge_col//返回当前难度模式下的列数
```

内部数组版: void play_game_base

```
bool row_col_select  
int complex_select_base  
void display_hide_array;//展示当前状态下的地图  
void DFS_display_unhide  
void display_unhide
```

伪图形界面版: void play_game_graph

```
void DFS_draw_unhide_graph  
void DFS_draw_unhide_graph  
void draw_unhide_graph//展示当前状态下的地图  
int graph_row_col_select
```

输出提示使用户开始游戏操作, complex_select_base/ graph_row_col_select 根据用户输入返回对应值, 如果退出返回 0, 输入合法下标返回 1, 查看时间返回 2, 标记/取消标记返回 3/4, 若用户操作不满足以上任何, 则直到操作为上述一种时才会结束该函数。程序根据返回的值进行下一步操作。0 则退出游戏返回菜单, 1 则调用 display_unhide/draw_unhide_graph 函数显示新的操作后的地图, 3/4 则标记/取消标记对应位置。每次操作后程序会调用 judge_game 判断游戏状态, 如果游戏胜利或失败返回对应提示, 否则游戏继续。