# 实习二 基于Floodfill算法实现扫雷游戏

## 一、实习目的与要求

**【问题描述】**

Windows扫雷是一款最经典的微软Windows系统自带的益智游戏，本次实习要求基于回溯算法设计并实现扫雷游戏。

**【游戏的操作方法】**

游戏主区域由很多个方格组成。使用鼠标随机点击一个方格，方格即被打开并显示出方格中的数字；方格中数字则表示其周围的8个方格隐藏了几颗雷；如果点开的格子为空白格，即其周围有0颗雷，则其周围格子自动打开。

例如，方格中出现1，说明上下左右及斜角合计有一颗雷，依次类推，2则有2颗，3则有3颗。在确实是地雷的方格上点了旗子，就安全了，不是地雷的被点了旗子，后面会被炸死。在不确定是否是地雷的方格上用右键标示"?"符号，表示怀疑这个格子是地雷。这样格子在自动打开时，被标示的格子就不会被打开，导致游戏结束。

在游戏中，当玩家不小心踩到地雷格子就算失败。而当地雷计数器中的数字变成0时，说明地雷全部被查到，游戏结束，玩家胜利。

【开发环境】

不限。可考虑MFC、QT、Java+SWT、C#、HTML5等。

#### 二、分析与设计

1. **需求分析**

需要根据上述的游戏规则模拟实现一个扫雷游戏，能够实现游戏中的基本功能。其中如果点击0时会向四周进行BFS搜索，即flood fill，这也是主要的算法。

1. **详细设计**

初始化：首先做好生成多个不同随机数的函数(类)。表示在扫雷网格范围内的雷的位置。之后开始做一个矩阵，并且进行初始化填充。首先初始化全为0，再将之前得到的雷的位置传入，将这些位置设为-1表示此处为雷。之后再将-1周围8个位置除雷的位置加一，这样就做好了初始化工作。得到了游戏的主要地图。

UI界面：做出游戏的主要窗口界面，用多个相邻的方形按钮模拟扫雷中的网格，用鼠标响应事件连接内部计算与用户的游戏操作。

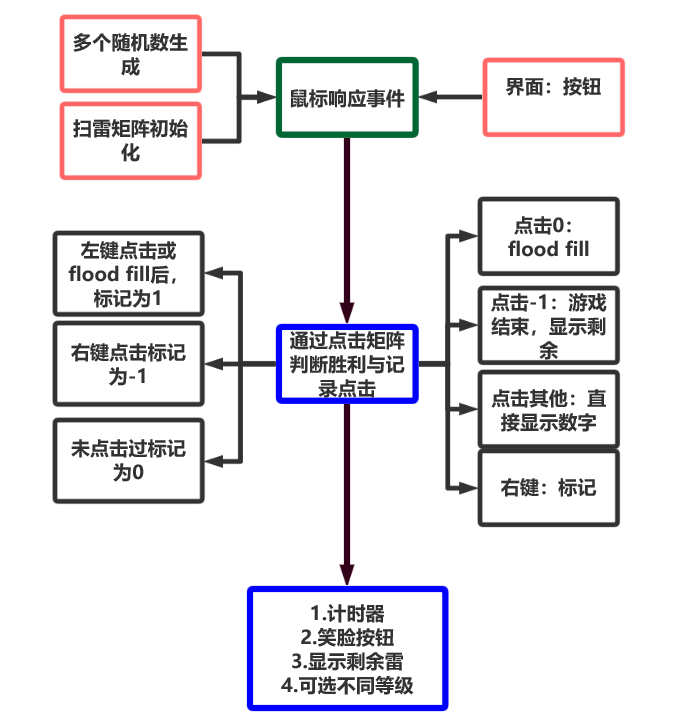


每一个按钮对应一个数字与点击状态。

鼠标响应事件：鼠标左键：如果鼠标点到0时则进行BFS搜索，向外拓展，直到遇到非0的数字停止拓展。如果是非零。直接显示它的数字，当然如果是点到-1(雷)时，则游戏结束并显示所有剩余的区域。鼠标右键：标记此处有雷，按钮显示图标，当然如果之前已标记则再点击右键则取消标记。

在初始化时已经有一个记录雷与数字的矩阵，之后还需要再设置一个等大的矩阵来记录鼠标点击的情况。用0表示未点击，用1表示已经被鼠标左键点击，或者是通过flood fill进行展开。标记后不能再进行点击或其他操作。用-1表示被鼠标右键点击过，当然再右键点击一次会变回0。这个矩阵主要是在按钮的鼠标响应时间内改变值。

完善：用多线程做一个计时器，如果游戏结束则停止。添加一个显示剩余雷的label，鼠标右键标记时会作出相应的加减操作。添加一个扫雷中的笑脸按钮，点击后可重新开始游戏。

****

1. **实现**
2. **出现的问题及解决办法**

（1）在多个鼠标响应事件设置时，需要设置final变量表明矩阵中具体的一位值。设置为final表明此值为常量，初始化一次之后不会改变，这与鼠标响应时间对应。

（2）点击笑脸重新开始游戏：将之前的构造函数里面的代码变为一个普通函数。构造时会调用，当然在点击笑脸时也会调用。这里可以利用java的垃圾处理来自动处理指针问题。

（3）其他问题：主要是在鼠标点击时，能够正确对两个矩阵进行处理与显示。

#### 【运行结果】



## 三、实习小结

本次实习算法还是比较简单的，主要是练习了java的界面的一些基本操作。通过实习，能够更加熟练地进行java编程，对图形化操作以及相关语句也更加熟悉。当然，本次实习还是存在一些不足的地方。对于一些临界条件的判断考虑不充分，对于插旗后的网格再flood后旗不会消失。。。对于更大的网格展开较慢，效率低。同时把主要算法全部写到按钮点击事件中逻辑较为混乱，且判断较多，代码不够精简。

## 四、课程学习总结及教学建议或意见

总结：

通过算法这门课的学习，我对算法有了很深的体会。算法类似一种简便方法，通过减少计算的时间来提高效率，而为了达到这一目标，我们写的代码必须更加精炼，对于问题的解决方式也绝非以暴力搜索而敷衍了事。我们需要花更多的时间来分析题目，找到适用于它的算法，并且根据具体问题进行一定的改进，之后再进行不断测试，然后学习那些耗时更短的算法，积累知识并提高自己的思维敏捷度。平时的课程主要是分为课上课下两部分，在课上主要是老师为我们讲解一些算法中的经典问题，来引出相关算法的思想。师傅领进门修行在个人，老师只是起到一个引领的作用，课上题目不需要讲的多么透彻，只需要点到为止，我们需要能够自己独立的思考问题，真正理解算法的思想。在课下主要是自己在leetcode上刷题，通过实战来锻炼自己的算法应用能力，通过每周的解题报告定期进行学习总结并了解自己的算法掌握程度。当然在刷题的这一过程中我体会到刷题不在于数量，而在于质量。每道题都需要能够深刻理解题目的内涵，同时学习他人更好的解题方法，对这两点进行结合，最后能够产出属于自己的知识与思想。总之，通过课上课下这两部分的学习，我确实掌握了许多算法知识，自己的解题思路与代码风格有所改变，对于解决问题的能力也有提高。

建议与意见：

1.课堂上尽量为同学们介绍更多的经典算法问题，主要是帮助同学们分析题目，并且给出几种解题方法，不用过于深入的去一步一步的讲解。能够让同学们课下，结合LeetCode上的练习，进行一定的总结与思考

2.对于软考题目的讲解，效率不是很高。可以进行一定的提炼，将题目变得更加直观，这样可以快速的讲解。总之是希望能够做出一定的改进，提高课堂效率。

3.可以适当的介绍一些前沿的知识与思想，帮助同学们开拓眼界。

4.每周的解题报告侧重点主要还是自己的体会与感受，对于本周的算法的理解与应用情况。能够将题目精做，结合discuss中的比较好的方法，来提升自己，而不是过于追求数量。