2019-2020学年第2学期

吉林大学《数据结构》课程设计报告

学院：软件学院 年级： 2018级 专业： 软件工程 班级： 1班

姓名：荆煜 学号： 55180123 题号：**A题**

1. 核心算法的设计思路和实现要点

（1）

1.为保证数组边界不越界，先将整个数组向右下方移动一格，即横纵坐标均加一。用数组a[][]保存游戏界面，b[][]保存全部挖开后的所有信息。

2.在埋入所有雷后，先遍历数组b生成含有相邻雷数目的完整信息。

3.使用递归挖雷，当挖到雷时失败，当挖到空白时，递归挖相邻8格中未挖开的格。

4.每次挖雷结束后，通过判断未挖格数与雷数是否相等来判断是否胜利。

（2）

1.同（1），编写递归程序dig()负责挖雷，编写程序panduan()判断是否胜利，并将（1）中其他部分整合入函数RefreshGamePanel()中。

2.为防止数组越界，将GamePanel[][]的所有数值向右下移动一格，存储于gamepanel[][]中。

3.判断是否已打开过，若未打开，则先挖右下角。

4.设置数组mine\_all[][]存储是否有雷，1为无雷，-1为有雷，0不确定。编写函数calculate()用于计算某点周围8个格中某特定数值的个数。编写函数clean()用于数组赋值。

5.遍历全图，若某点周围未挖开的格数等于自身数字，或周围确定无雷的格数与未挖开格数之和为8，则未挖开格均为雷，标记于mine\_all()中；若某格周围确定有雷格数等于该格格数，则其余格均为无雷，标记于mine\_all()中；创建数组mark[]标记是否可能有雷，编写函数open()，若某点周围某格未打开且mine\_all[][]不能排除无雷，则标记该点并递归执行，count存储标记次数。

6.遍历，若某格mine\_all标记无雷且未打开，则点击该格。优先打开标记点数最多的格。若全部确定无雷格全部打开且游戏未胜利，则计算不确定是否有雷格所能标记数，标记越多，则该点无雷可能越大，打开该格。

7.若仍无法确定，则从左上角开始连续点击。

1. 算法时间、空间复杂度的估算
2. 完成过程中遇到的问题，排除问题的主要过程、使用的方法和技巧

(1)

1.未考虑多次点击已打开格的情况。增加点击时的判断。

(2)

1.移植(1)问时，因题目中mine[][]为是否标雷，与第一问自己定义不符。设置临时变量解决。

2.某格翻开为n，周围有m格未翻开，若将周围点是雷的概率定为n/m，则失败率极高。

2.在无法找到下一个确切无雷格时该点击哪一格。对于已知雷情况进行递归遍历，从某点呢周围开始标记所有可能为雷的点，标记点数越多，该点为雷的几率越小。

1. 所用方法的特别、新颖或创新之处
2. 通过向右下移动一个以减少防止数组越界而作出的判断次数。
3. 通过使用mark来判断可能性而非通过等概率判断。