Lba3 扫雷初级版本

一 、程序主要步骤

1. 创建Board3类的一个对象newGame,读入地图长宽及地雷数, 生成地雷图及游戏界面。

2. 使用 while 循环不断读入用户输入的坐标以进行游戏,并用 Board3类中click方法检测是否未有效坐标。 若不是，提醒用户并重新输入。

3. 读入用户输入的有效坐标后,用 if 语句检测指定坐标是否为地雷。如果是地雷,则打印地雷 图并提示用户”踩中地雷,游戏结束! ”并结束程序;如果不是将初始地图中指定坐标处的元素赋 值为地雷图中相同坐标的元素,并打印赋值后的初始图,继续循环让用户输入下一个坐标。

4. 最终扫雷成功的判断(while 循环的条件):设置一计数器 i 以记录正确输入坐标的次数,初始 化为 0 ,计算扫雷成功需要正确输入坐标的次数,并赋值给整型 变量 rightClick 。每次正确输入坐标后计数器 i 加一。当循环正常结束时,判定为扫雷成功 。用 if 语句在 while 循环体之后判断扫雷是否成功,若成功,提示用户”恭喜你,扫雷完成! ”。

二 、各项功能的实现

1. 打印初始游戏界面:

(1) 创建数组:用户输入长宽及地雷数后,set方法将其设置为对象newGame的私有数据,并生成三个数组,布尔类型 state 检测地雷,字符类型 bombLable 作地雷图(地雷图生成的具体实现以在Lab2中讨论),字符类型 gameInterface 作游戏界面。嵌套 for 循环为 gameInterface 数组赋初值,元素全为字符#。

(2) printGameInterface 方法用嵌套for 循环将gameInterface数组元素顺序按格式打印。

2. 输入坐标后的反馈:

(1) 读入坐标

click方法检测输入合法性。

a. 未按指定格式输入

用正则表达式检测是否为规定格式。

注意:coordinate 读入用户输入坐标时使用了 nextLine,为防止程序开始输入长宽及地雷数后按下 Enter 键对坐标输入造成影响,在 while 循环体之前添加 String tab = cmd.nextLine()。

b. 不在地图内

用 if 语句判断是否超出,若超出,提示用户”Not whthin the map,click again! ”。

c. 重复输入

if 语句判断输入坐标在gameInterface 数组对应元素是否为#,若不是,则当前坐标已被点击,提示用户”You have clicked it,click others! ”。

注意:测试顺序必须先 b 后 c,否则若在检测 c 情况时,坐标点不在地图内时(即 b 情况),无法在gameInterface 数组中找到对应元素, 出现错误。

(2) 反馈

情况分地雷和非地雷两种。if 语句,布尔数组 state 判断是否为地雷,若是, printBomb Lable 方法打印地雷图,提示用户”踩中地雷, 游 戏 结 束 ! ”,并 结 束 程 序 ; 若 不 是 , 将 地 雷 图 中 相 应 坐 标 的 元 素 赋 值 给 游 戏 界 面 , 打印游戏界面。计数器i加一,若i小于rightClick,提示用户”Good job,go on!”。将Boara类中布尔类型数据isVictory赋值为i == rightClick。

3. 判断扫雷成功

while循环条件设置为(!isVictory),循环正常结束则提示用户”恭喜你,扫雷完成! ”。

1. 对象和类

据实验要求定义一个Board3类,并创建一个newGame对象。以下为UML类图。

Board3

- rows : int

- cols : int

- bombs : int

- rightClick : int

- i : int

- bombLable : char[][]

- state : boolean[][]

- gameInterface : char[][]

+ isValidClick : boolean

+ isVictory : boolean

+Board3()

+setArgument(rows:int,cols:int,bombs:int):void

+creatArray():void

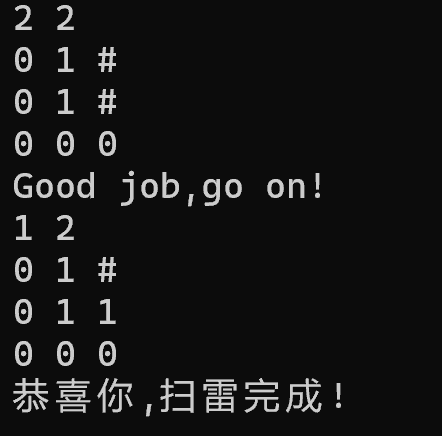
-printBombLable():void

-printGameInterface():void

+click(s:String):void

四、运行效果截图

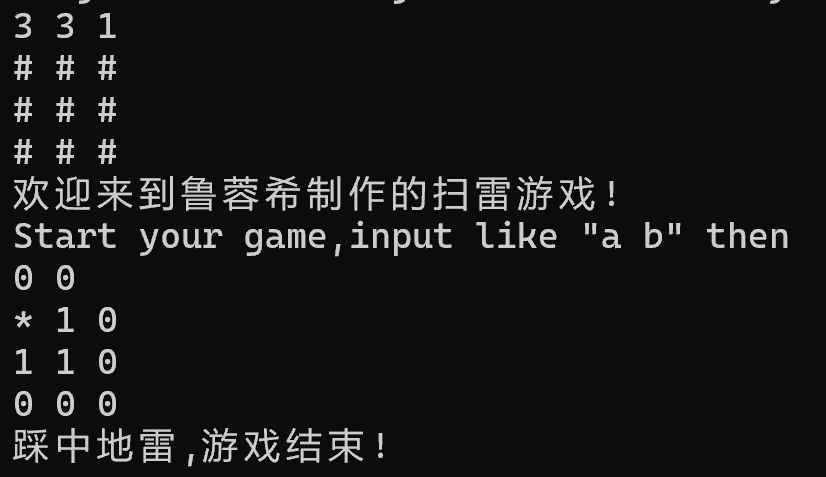
Case 1:扫雷完成



Case 2:未按格式输入坐标



Case 3:踩雷,游戏失败



附录:

关键代码

1. while 循环

**while**(!newGame.isVictory) {

String coordinate = cmd.nextLine();

newGame.click(coordinate);

}

2. click方法

**public** **void** click(String s) {

isValidClick = **true**;

**int** r = 0;

**int** c = 0;

**if**(s.matches("\\d+\\s\\d+")) {

Pattern pattern = Pattern.*compile*("(\\d+)\\s(\\d+)");

Matcher matcher = pattern.matcher(s);

**if** (matcher.find()) {

r = Integer.*parseInt*(matcher.group(1));

c = Integer.*parseInt*(matcher.group(2));

}

}

**else** {

System.***out***.println("Not a click,try again!");

isValidClick = **false**;

}

**if**(r >= rows | c >= cols) {

System.***out***.println("Not within the map,click again!");

isValidClick = **false**;

**return**;

}

**else** **if**(gameInterface[r][c] != '#') {

System.***out***.println("You have clicked it,click others!");

isValidClick = **false**;

**return**;

}

**else** **if**(state[r][c]) {

printBombLable();

System.***out***.println("踩中地雷,游戏结束!");

System.*exit*(1);

}

**else** {

gameInterface[r][c] = bombLable[r][c];

printGameInterface();

i++;

isVictory = (i == rightClick);

**if**(!isVictory)

System.***out***.println("Good job,go on!");

}

}