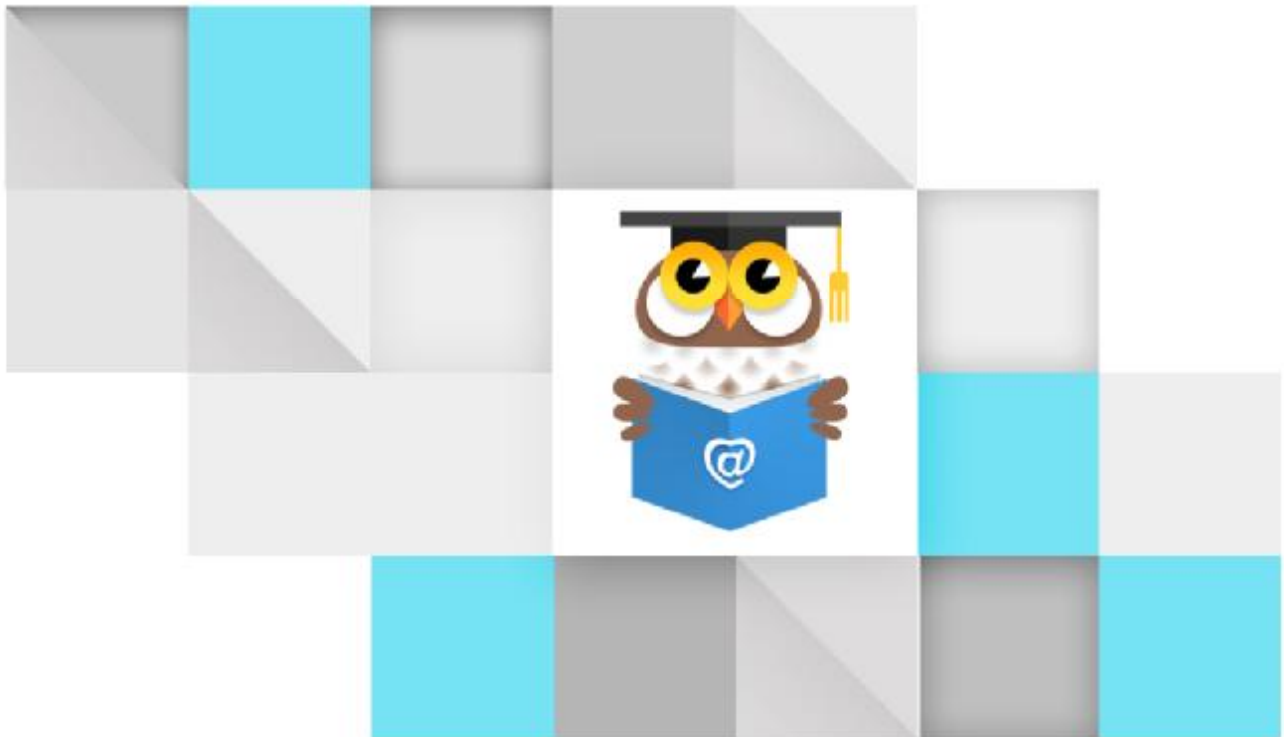


消灭泡泡糖 (Java)

实训指导手册

实训场景 013 – 移动水平方向的泡泡糖



Campus Solution Group

目 录

一、任务编号：PRJ-BU2-JAVA-013	1
1、实训技能	1
2、涉及知识点	1
3、实现效果	1
4、场景说明	2
5、快速开始	4
6、任务 1 – 获取被整列清空的泡泡糖集合	4
7、任务 2 – 获取水平待移动泡泡糖	6
9、场景总结	9

一、任务编号：PRJ-BU2-JAVA-013

1、实训技能

I Java API 运用技能

2、涉及知识点

I List 中的方法

I List 的应用

I 泛型类和泛型接口

3、实现效果



图 3-1

4、场景说明

1、业务说明：

本场景主要实现了在整列泡泡糖全部消除后，右侧泡泡糖向左侧水平移动的功能。

2、实现思路：

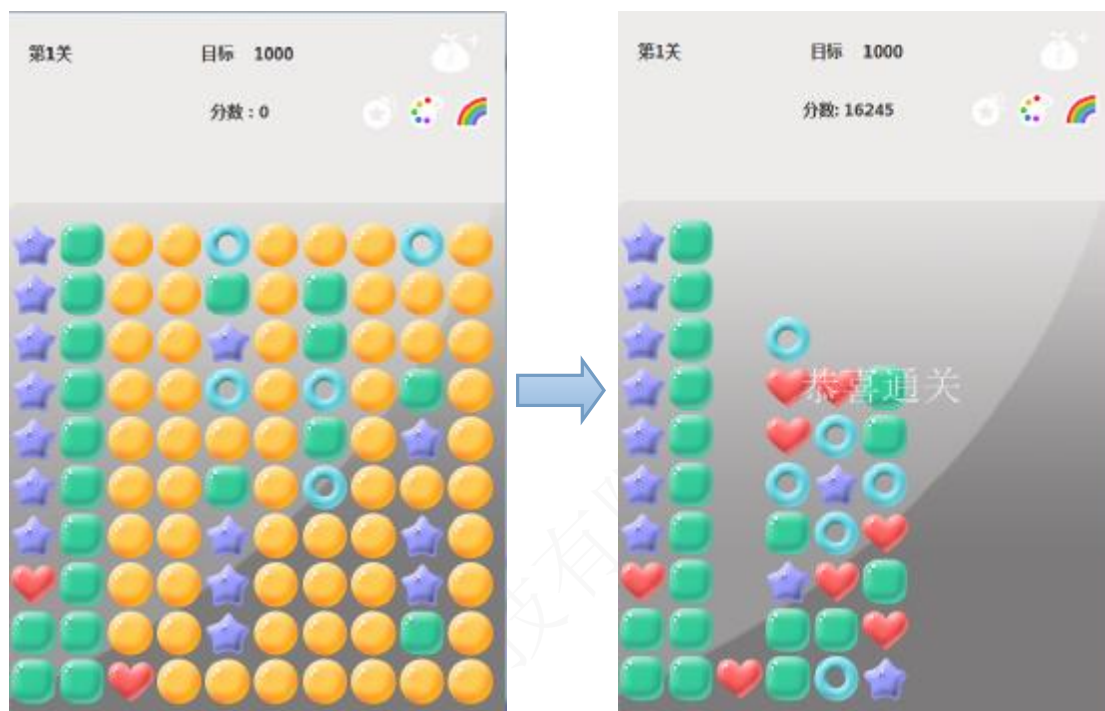


图4-1

由图4-1中可见，当用户点击某个黄色泡泡糖，游戏中完成以下操作：

2-1. 【界面】消除相同颜色的泡泡糖

2-1.1. 【业务】获取待消除泡泡糖返回给界面 - `tobeClearedStars`函数

2-2. 【界面】泡泡糖发生自顶向下的垂直移动“填补空缺”。

2-2.1. 【业务】获取垂直方向待移动泡泡糖返回给界面 - `getYMovedStars`函数

2-3. 【界面】泡泡糖发生从左向右的水平移动“填补空缺”（如需要）。

2-3.1. 【业务】获取水平方向待消除泡泡糖返回给界面 - `getXMovedStars`函数

3、核心组件介绍：

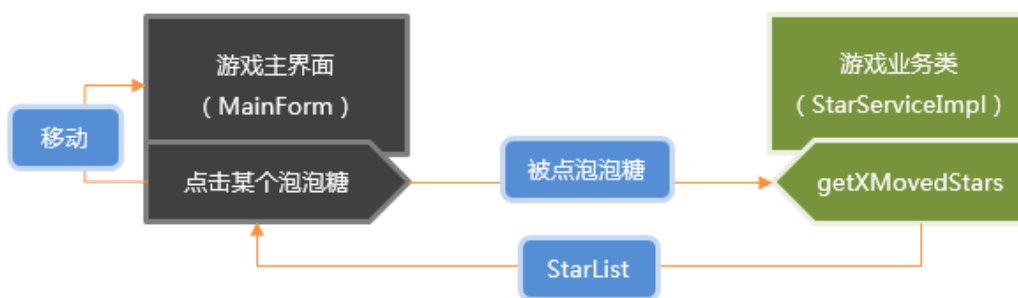


图4-2

3-1. MainForm - 游戏界面类（本场景无需实现）：

负责游戏数据显示、响应用户在界面上的各类操作。

3-2. StarServiceImpl - 游戏业务类

负责游戏相关逻辑计算，例如：泡泡糖移动、消除、分数计算等操作。

3-3. StarList - 用于保存待移动泡泡糖（MovedStar）集合。

游戏界面会根据getXMovedStar返回的待移动泡泡糖集合（StarList）实现泡泡糖水
平移动操作。

3-4. StarServiceImpl类中的getNullColumns：

该方法主要负责获取“被清空所有泡泡糖的列集合”，该方法会在任务1中实现。

3-5. StarServiceImpl类中的getXMovedStars：

该方法主要基于任务1实现的功能函数，计算“水平方向待移动泡泡糖”的数量和移动
步长，并保存在StarList中。

4、了解更多：

请参考《消灭泡泡糖 - 需求说明文档》

5、前置条件：

5-1. 前置场景：PRJ-BU2-JAVA-012 - 移动垂直方向的泡泡糖（三）

5-2. 必备知识与技能：

5-2.1. Java开发工具（Eclipse）。

5-2.2. Java面向对象编程技能（嵌套for循环，if条件块，类的成员方法）。

5、快速开始

1、开发环境：

1-1. Oracle JDK8.x 以上版本

1-2. Eclipse Luna（4.4.x）以上版本

1-3. 工程包：PRJ_BU2_JAVA_013

2、进入开发环境：

详见SPOC平台上《PRJ-BU2-JAVA-013 前置任务：进入开发环境》



图 5-1

6、任务 1 – 获取被整列清空的泡泡糖集合

1、任务描述：

1-1. 当前任务主要是根据界面泡泡糖矩阵，获取“被清空所有泡泡糖的列”集合。

1-2. 由场景说明可知，在水平移动之前已经完成了垂直方向泡泡糖移动，因此我们可以通过对某列泡泡糖“最底部成员”是否存在，来判断该列泡泡糖是否“已被全部清空”。

2、推荐步骤：

2-1. 场景定位

2-1.1. 定位到：cn.campsg.practical.bubble.service.StarServiceImpl类

2-2. 创建私有函数getNullColumns

2-2.1. 函数参数：当前界面泡泡糖集合

2-2.2. 返回类型：整数类型的集合，集合中的每个元素代表一个空列的列号。

2-3. 创建整数类型的集合实例，用于存放“被清空所有泡泡糖的列”数据。

2-4. 按列号遍历整个泡泡糖集合，判断并获取“被清空所有泡泡糖的列值”。

2-4.1. 并判断每列“最底部泡泡糖”是否存在？

2-4.2. 如果为null，表示当列“最底部泡泡糖”不存在，将列值加入步骤2-3定义的集合中。

+ 提示：

- 1) 矩阵的坐标是从0开始的，因此遍历范围为：【0】~【最大列值-1】。
- 2) 搜索某个位置上的泡泡糖可以利用StarList的lookup方法。
- 3) “最底部泡泡糖”，就是行号 = MAX_COLUMN_SIZE - 1。
- 4) 最大列值MAX_COLUMN_SIZE可从StarService的常量中获取。
- 5) 最大行值MAX_ROW_SIZE可从StarService的常量中获取。

2-5. 方法返回该整数类型的集合

3、验证与测试：

3-1. 定位到StarServiceImpl类的getXMovedStars函数

3-2. 添加测试代码

3-2.1. 调用getNullColumns函数，将getXMovedStars方法的参数传入

getNullColumns函数，获取空列集合。

3-3. 打印测试结果：

3-3.1. 在getXMovedStars函数中打印出空列集合。

3-3.2. 运行项目工程，选择启动函数：cn.campsg.practical.bubble.MainClass

3-3.3. 请在界面上点击任意“黄色泡泡糖”，输出结果请与下图一致：

[3, 5, 7, 9]

图 6-1

7、任务 2 – 获取水平待移动泡泡糖

1、任务描述：

本任务主要利用【任务1】计算获得的“整列被清空泡泡糖”的列号集合，获取水平待移动的泡泡糖的数量和移动步长，实现方法主要分为以下几步：

- 1) 首先，获得【任务1】计算所得的“整列被清空泡泡糖”的列号集合。
- 2) 然后对集合进行遍历：
 - a) 遍历起始位置：列号集合第一个元素，即第一个空列。
 - b) 遍历结束位置：最后一列。
 - c) 每次遍历执行：判断当前列是否属于空列：如果是，则水平移动步长+1，如果不是，则将整列泡泡糖依次封装为“待移动泡泡糖”（需要移动）。
- 3) 最后返回待移动的泡泡糖集合。

2、推荐步骤：

2-1. 场景定位

2-1.1. 定位到：cn.campsg.practical.bubble.service.StarServiceImpl类

getXMovedStars函数

2-1.2. 找到注释：/** PRJ-BU2-JAVA-013 Task2 【2/2 start】 */ 处

2-1.3. 删除测试用的打印代码。

2-2. 调用getNullColumns获取空列集合，参数-泡泡糖集合可用getXMovedStars的入参。

2-3. 如果空列集合不存在，则表示X轴无需移动，返回null结束getXMovedStars函数。

+ 提示：

1) 空列集合可以调用任务1中的getNullColumns函数获取。

2-4. 定义初始化变量

2-4.1. 创建“待移动泡泡糖”集合：该对象用于保存所有水平方向移动的泡泡糖。

2-4.2. 创建X轴移动步长变量：该变量存储水平方向泡泡糖移动的距离，初始值为0。

2-5. 依次获取列集合中的成员，根据空列列号做如下判断操作：

2-5.1. 如果是空列，则让X轴移动步长变量值+1，并进入下一次循环。

2-5.2. 如果不是空列，那该列中的泡泡糖就是需要移动的，进入保存待移动泡泡糖步骤。

+ 提示：

1) 循环判断所有空列的范围：空列集合第一个成员 ~ 游戏的最大列号。

2) 判断是否为空列：利用集合的contains函数判断当前循环变量是否在空列集合中。

2-6. 保存待移动泡泡糖的逻辑如下：

2-6.1. 从底向上依次获取非空列中的所有泡泡糖。

2-6.2. 判断获取的泡泡糖，如为null表示集合顶部已无泡泡糖，直接中断循环。

2-6.3. 如不为null，则利用StarsUtil的静态方法toMovedStar将“泡泡糖类型”转换成“待移动泡泡糖类型”。

+ 提示：

1) 从底向上依次获取非空列中所有泡泡糖的判断范围：【最大行号常量-1】 ~ 0

+ 业务说明：

1) toMovedStar函数实现了将“泡泡糖类型”Star转换成“待移动泡泡糖类型”MovedStar，的操作，即将Star类型中的position和type属性复制给MovedStar。

2) 源代码如下：

```
public static MovedStar toMovedStar(Star star) {  
    MovedStar ret = new MovedStar();  
    ret.setPosition(new Position(star.getPosition().getRow(), star  
        .getPosition().getColumn()));  
    ret.setType(star.getType());  
    return ret;  
}
```

2-6.4. 设置待移动泡泡糖的待移动位置属性【MovedPosition】的行、列值

1) 行：等同于泡泡糖的行值（水平移动无需考虑行值的变化）。

2) 列：泡泡糖的原始列值 - 需要移动的步长。

2-6.5. 把【待移动泡泡糖】添加到【待移动泡泡糖集合】中

+ 业务说明：

1) 与垂直方向泡泡糖移动不同，水平移动距离 = 泡泡糖的原始列值 - 移动步长，使用减法（而非加法）的原因在于，水平移动是从左向右的移动操作，列号趋于减小。

2-7. 将方法的返回值从null修改为“待移动泡泡糖”集合。

3、验证与测试：

3-1. 运行项目工程，选择启动函数：cn.campsg.practical.bubble.MainClass

3-2. 点击任意“黄色泡泡糖”，测试Y和X轴泡泡糖移动功能：

3-3. 输出结果与下图一致：



图 7-1

9、场景总结

无