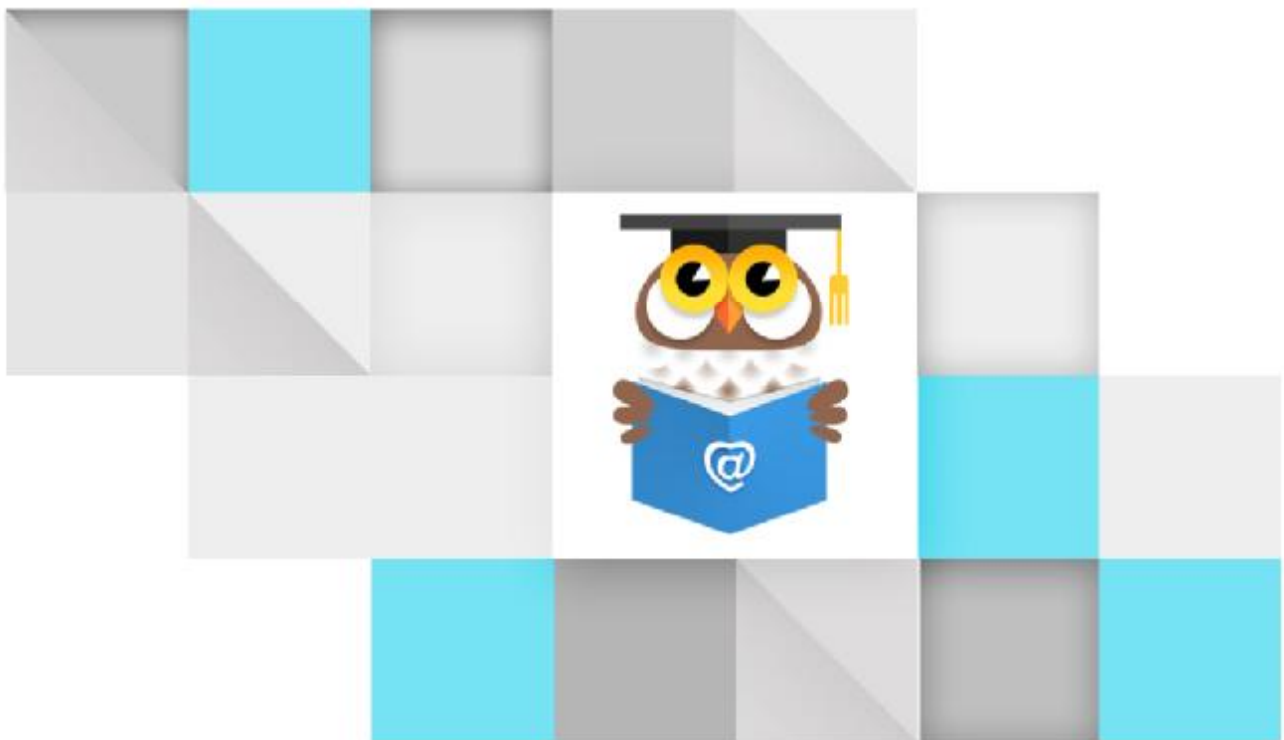


消灭泡泡糖 (Java)

实训指导手册

实训场景 014 – 更新关卡通关分数



Campus Solution Group

目 录

一、任务编号：PRJ-BU2-JAVA-014	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	1
5、快速开始	4
6、任务 1 – 获取初始通关分数	5
7、任务 2 – 更新关卡通关分数	8
8、场景总结	10

一、任务编号：PRJ-BU2-JAVA-014

1、实训技能

I Java API 运用技能

2、涉及知识点

I IO 流相关类设计

I 读文本数据

3、实现效果



图 3-1

4、场景说明

1、业务说明：

1-1. 本场景主要实现了通关后，下一关卡通关分数的获取与计算。

1-2. 本场景将利用IO流技术对文件执行读取操作，并对获取的分数执行业务操作。

1-3. 所有与游戏分数有关的数据都保存在配置文件（score.conf）中。

2、实现思路：

2-1. 为保证游戏的可维护性，我们将游戏分数相关信息保存在配置文件（score.conf）中。

2-2. 该配置文件由Configuration组件负责读取，并将获取的通关分数保存于Score类中。

2-3. ScoreServiceImpl根据游戏业务从Score实体中获取分数信息，并根据以下场景执行各类业务计算：

2-3.1. 游戏初始化时，ScoreServiceImpl对象负责获取第一关的通关目标分数。

2-3.2. 游戏通关时，ScoreServiceImpl负责计算下一关通关目标分并显示。

3、核心组件介绍：

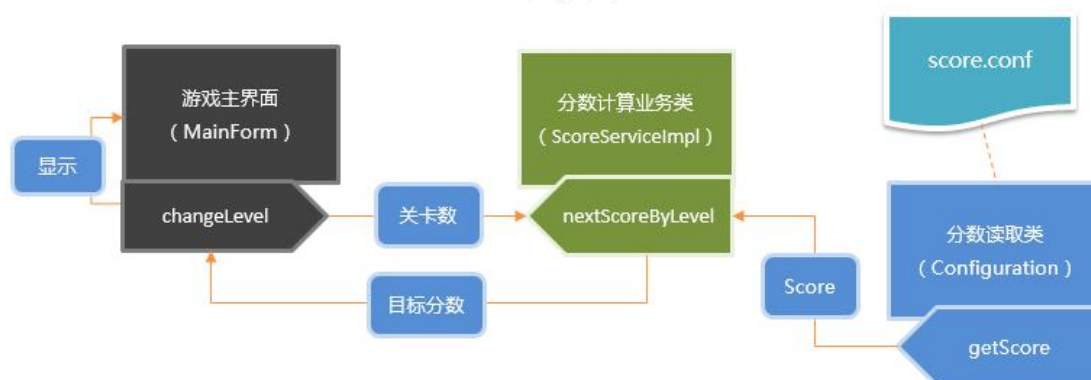


图 4-1

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. ScoreServiceImpl – 积分业务类

负责积分相关逻辑计算，例如：通关分数、消除得分、结算奖励等操作。

3-3. Score

3-3.1. 用于保存配置文件中的游戏分数信息。

3-3.2. 为了使游戏难度呈递增趋势，我们将配置文件数据分为四个部分：

- 1) 每关的通关目标分 - levelScore：默认第一关，之后随关卡变化而更新；
- 2) 每关的通关递增值 - step：该分数表示关卡切换时，每关通关目标分递增值。

例如：

第一关：目标分：1000分

第二关：目标分：3000分 = 第一关目标分 (1000) + 通关递增值 (2000)

第三关：目标分：5000分 = 第二关目标分 (3000) + 通关递增值 (2000)

- 3) 难度追加递增值 - increment：该分值每隔固定关数会增加一倍，属于控制游戏难度递增的重要参数。

例如1：

第四关：目标分：7000分 = 第三关目标分 (5000) + 通关递增值 (2000)

第五关：目标分：10000分 = 第四关目标分 (7000) + 通关递增值 (2000) + 难度追加递增值 (1000)

例如2：

第八关：目标分：19000分 = 第七关目标分 (16000) + 通关递增值 (2000) + 难度追加递增值 (1000)

第九关：目标分：23000分 = 第八关目标分 (19000) + 通关递增值 (2000) + 2 * 难度追加递增值 (1000)

- 4) 难度追加关卡间隔数：length，该数值表示每隔几关，追加分翻倍。

例如：

难度追加关卡间隔数 = 4；

难度追加递增分 = 1000

那么：

01~04关：0，无难度追加递增分

05~08关：1000，增加难度追加递增分

09~12关：2000，增加难度追加递增分 * 2

13~16关：3000，增加难度追加递增分 * 3

以此类推。

3-4. Configuration：

该类作为得分配置文件的读取类，主要的作用是，读取配置文件中的得分信息，封装到Score中，以便在ScoreServiceImpl中进行相关业务计算。

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-013 – 移动水平方向的泡泡糖

5-2. 必备知识与技能：

5-1.1. Java开发工具（Eclipse）。

5-1.2. Java面向对象编程（算术运算、类的成员变量、类的成员方法、构造器的调用、捕捉多个异常）。

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_014

2、进入开发环境：

详见SPOC平台上《PRJ-BU2-JAVA-014 前置任务：进入开发环境》



图 5-1

6、任务 1 – 获取初始通关分数

1、任务描述：

当前场景任务是，获取第一关的通关目标分数，该分数需要从配置文件中动态读取。

2、推荐步骤：

2-1. 场景定位

2-1.1. 定位包：cn.campsg.practical.bubble.service

2-2. 创建用于读取积分配置文件的类 - Configuration

2-3. 添加配置文件路径的常量

+ 业务说明：

因配置文件score.conf在项目src根目录下，所以设置相对路径为："score.conf"。

2-4. 创建Score成员变量，用于保存所有从配置文件中读取的游戏分数数据。

2-5. 创建公共函数，用于返回2-4步骤创建的Score成员变量（建议：getScore方法）。

2-6. 创建Configuration类的0参构造器，用于读取文件数据并封装到Score成员变量中，在构造器中完成步骤2-7、2-8、2-9：

2-7. 实例化Score，为后续存放读取的数据做准备。

2-8. 利用java.io流包下的字符流对象，读取配置文件

2-8.1. 创建字符流缓存读取器（BufferedReader）读取配置文件

+ 提示：

获取文件的缓存读取器（BufferedReader）步骤如下：

1) 读取路径定位src目录：

通过getClass().getClassLoader()对象的getResourceAsStream方法可直接定位当前工程的src目录，返回的InputStream对象定位的根目录就是src。

`getClass().getClassLoader().getResourceAsStream(配置文件路径的常量)`

2) 利用java.io.InputStreamReader将InputStream转换成BufferedReader对象。

`new BufferedReader(new InputStreamReader(is)); //is为InputStream对象。`

2-8.2. 读取所有泡泡糖的分数数据并保存入Score对象中。

+ 提示：

1) 打开score.conf文件，观察数据为四行整型数据，依次为：

1-1) 第一行：每关的通关目标分，数据需保存入levelScore属性

1-2) 第二行：每关的通关递增分，数据需保存入step属性

1-3) 第三行：难度追加递增分，数据需保存入increment属性

1-4) 第四行：难度追加关卡间隔数，数据需保存入length属性

2) 利用BufferedReader的readLine函数依次获取每一行的数据。

2-8.3. 关闭文件的缓存读取器，释放文件资源。

2-9. 抓取读取文件中可能遇到的异常

2-9.1. 请分别捕捉FileNotFoundException、IOException。

2-9.2. 如发生以上两个异常，请将score置空，表示读取失败。

2-10. 找到cn.campsg.practical.bubble.service.ScoreServiceImpl类

2-11. 定位：getCurrentLevelScore方法

2-12. 获取并返回第一关初始目标分：

2-12.1. 首先，在ScoreServiceImpl类中创建Configuration成员变量，初始值为null。

2-12.2. 其次，编写ScoreServiceImpl的0参构造器，实例化Configuration成员变量。

2-12.3. 最后，定位getCurrentLevelScore方法，返回Configuration中Score属性的每关的通关目标分 - levelScore属性。

3、验证与测试：

3-1. 运行项目工程的MainClass函数

3-2. 在界面顶部菜单显示：第一关分数为1000。

3-3. 该数据与score.conf第一行数据相同，见以下完成效果：



图 6-1

7、任务 2 – 更新关卡通关分数

1、任务描述：

本任务主要负责实现通关后，计算与显示下一关的通关分数，并更新Score中levelScore属性，保证该属性永远为最新关卡的目标分数。

每一关的通关目标分计算规则如下：

1-1. 下一关通关分数 (levelScore) =

当前通关分 (levelScore) + 每关通关递增分 (step) + 难度追加分 (x)。

1-2. 难度追加分 (x) =

(下一关卡 - 1) / 难度追加关卡间隔数 (length) * 难度追加递增分 (increment)。

1-3. 样本数据如下：

关卡数	下一关通关分 (levelScore)	当前通关分 (levelScore)	通关递增分 (step)	难度追加分 (x) (下一关卡 - 1) 整除 length * increment
1	3000	1000	2000	(2-1) / 4 * 1000 = 0
2	5000	3000	2000	(3-1) / 4 * 1000 = 0
3	7000	5000	2000	(4-1) / 4 * 1000 = 0
4	10000	7000	2000	(5-1) / 4 * 1000 = 0
5	13000	10000	2000	(6-1) / 4 * 1000 = 1000
6	16000	13000	2000	(7-1) / 4 * 1000 = 1000
7	19000	16000	2000	(8-1) / 4 * 1000 = 1000
8	23000	19000	2000	(9-1) / 4 * 1000 = 2000

2、推荐步骤：

2-1. 场景定位：

2-1.1. 定位到类：cn.campsg.practical.bubble.service.ScoreServiceImpl

2-1.2. 定位函数：nextLevelScore

2-2. 编写更新与获取通关分数方法：nextLevelScore

2-2.1. 方法参数：下一关关卡号。

2-2.2. 方法返回值：下一关通关分数（即通关后的关卡分数）。

2-3. 按任务说明中《每一关的通关目标分计算规则》计算下一关的目标分。

2-4. 用计算结果更新Score的通关分数（levelScore）。

2-5. 返回当前通关分数。

3、验证与测试：

3-1. 找到cn.campsg.practical.bubble.service.ScoreServiceImpl类

3-2. 测试结果：

3-2.1. 运行项目工程的MainClass函数，看到全屏一色泡泡糖。

3-2.2. 点击任意泡泡糖便能实现通关。

3-2.3. 请至少完成9关的通关操作，观察界面【目标】数值的变化情况如下：

1000 à 3000 à 5000 à 7000 à 10000 à 13000 à 16000 à 19000 à 23000



图 7-1

8、场景总结

Q1. 如何获取工程src目录下的文件？

1. 在Java中读取工程src目录下的文件 ,不该直接使用"src/xxx.xx"这样的路径 ,理由是Java程序发布后在不同操作系统中 ,系统对于"src/xxx.xx"路径理解是不同的。

2. 正确的读取方法是：

`getClass().getClassLoader().getResourceAsStream("src/xxx.xx")`。

以上方法会以虚拟机（JVM）作为相对路径映射"src/xxx.xx"。

* 对于初学者而言，以上内容只要牢记和熟练使用即可。

Q2. 读取本地文件您会选择哪个组件？为什么？

1. `BufferedReader`，该组件是读取本地文件的最佳选择。

2. `BufferedReader`类中的`readLine`可按行读取文件,返回`null`时表示读取完成,非常方便。

Q3. 字节流与字符流的区别是什么? 分别在哪里应用?

1. 字节流按字节读取数据,适用于网络数据交换,读取本地或网络的二进制文件(图片、视频等)。

2. 字符流按字符读取数据,适用于读取文本字符文件。

扩展: `InputStreamReader`和`OutputStreamWriter`是两个非常重要的组件,它们可以帮助程序员将字节流转换为字符流。

例如:

```
BufferedReader reader = new BufferedReader(new InputStreamReader(  
    new FileInputStream("C:/1.txt")));
```