

《连连看》Java课程设计方案

@author 施嘉宏 @startDate 2020.11.20 @lastModifiedDate 2020.11.26

目录

《连连看》Java课程设计方案

目录

Swing界面组成

需求分析

- 1.核心界面
- 2.信息栏
- 3.菜单栏
- 4.关于界面
- 5.欢迎界面
- 6.结算界面

界面结构图

类的封装

GameClient
EntryPanel
OverPanel
MessagePanel
GamePanel
Block
BlockFactory
AboutDialog

游戏逻辑构建

需求分析

- 1.记录方块分布数据
- 2.操作方块分布数据
- 3.生成方块分布数据

核心算法

- 1.直接连通
- 2.单拐点连通
- 3.双拐点连通
- 4.寻找连通点

方案一

方案二

- 5.数据随机

判断流程图

连通示意图

类的封装

LinkType
GameMap
LinkResult

Swing界面组成

需求分析

1.核心界面

位于窗口中央，核心界面由 $n*n$ 的方块组成，同种方块成对出现，并且是乱序的。每个方块一张图片，同类方块图片相同。方块间留足间隙满足观赏性。方块点击后应有方框标记。

2.信息栏

位于窗口上方，提示栏包括游戏的标题和游戏的具体信息，包括游戏剩余时间(s)，时间进度条(%)和得分情况。

得分：根据时间、重置次数、消去方块得出。其中消去方块分最多为 $(n*10)$ 分，在此基础上除以(消耗时间*重置次数*0.1)。

3.菜单栏

位于窗口最上方横条，菜单栏应包含游戏暂停、游戏重开、游戏退出、方块再次打乱、提示方块和“关于”界面的选项，应使用菜单和子菜单组合。游戏退出后显示欢迎界面。

4.关于界面

新的对话框，由菜单栏启动，内容应包含该项目的基本信息，包括作者、时间、项目名称、源码地址等。

5.欢迎界面

可切换的界面，用于打开主界面。界面中应包括游戏标题、开始按钮、关于按钮和退出按钮。开始按钮点击后应显示难度选择按钮。

6.结算界面

可切换的界面，显示得分和总消耗时间。包括退出、重开按钮。退出后显示欢迎界面。

界面结构图



类的封装

GameClient

extends JFrame

界面程序的本体，用于联络各个组件，控制游戏开始结束。

功能：

- 创建菜单栏、信息栏、游戏界面、关于界面、欢迎界面、结算界面。
- 订阅组件的监听器，实现各种页面的切换。页面切换使用CardLayout实现。
- 开始游戏和重开游戏，设定各个组件的初始参数。

EntryPanel

extends JPanel

退出或者开始。记录选择的难度（地图规模、方块类数）。

功能：

- 提供获取所选难度的方法
- 切换界面>(GamePanel+MessagePanel)

OverPanel

extends JPanel

结算界面，显示成功或者失败信息。

功能：

- 重开游戏>GameClient(reopen)
- 切换界面>EntryPanel
- 获取从MessagePanel中获取游戏信息

MessagePanel

extends JPanel

显示信息栏界面的内容。主要用于记录分数和时间。

功能：

- 各种参数的getter/setter方法。
- 当时间耗尽时调用自定义监听器

GamePanel

extends JPanel

使用Block对象填充GridLayout。根据GameMap对象提供的数据绘制Block，方块相消判定成功后绘制连线，记录数量。

功能：

- 为每一个Block添加MouseClicked监听器：被点击时传递自身的引用给GamePanel。
- 使用GameMap判断两个坐标是否能连通后，若成功则画线并检测是否消除完全，失败则改变所选取的Block状态，并删除其引用。
- 画线：

1. 直接连通：直接使用两方块坐标画线
 2. 单拐点：两方块坐标以及一个拐点，用三个点画线段。
 3. 双拐点：两方块坐标以及两个拐点，用四个点画线段，应注意线段总是在X或Y轴方向上。
- 刷新GameMap后，遍历Blocks，重置Blocks状态
 - 提示两个可以相消的方块。
 - 当方块完全消除后调用自定义监听器。
 - 当消除一次方块后调用自定义监听器。

Block

extends JComponent

保存方块在Map上的下标。

功能：

- 重置方块，包括图片、ID、状态（利用BlockFactory）
- 被选取时，绘制方框
- id的getter

BlockFactory

单例工厂类，用enum类型。

功能：

- 通过方块的id，生成一个带图片的Block对象。
- 通过方块id，重置一个现成的Block对象。

AboutDialog

extends JDialog

显示关于该游戏以及该项目的信息。

游戏逻辑构建

游戏逻辑主要在于方块数据处理和判断两个方块是否能消去。

需求分析

假设地图规模大小为 $n*n$ ，方块种类为 i 种

1.记录方块分布数据

一个二维数组，不同的ID对应不同的方块，同一个ID应成对出现。空白方块使用特殊标记。

2.操作方块分布数据

重新打乱二维数组中的数据，应注意只能打乱非空白方块的数据，空白的位置应保持不变。

判断两个方块是否能够相消，以何种方式相消。

遍历查找两个可连通的点。

传出和修改对应点的方块ID。

3.生成方块分布数据

根据地图大小，创建一个二维数组，并在里面随机添加方块的ID。每种类型的方块添加 $(n*n)/i$ 个，共 $(n*n)/2i$ 组。根据 n 的不同， i 需要改变为 n 的最大约数（且小于等于 i 的最大值）。

核心算法

按照连连看规则——可连接的两个方块之间最多用三条直线相连——可以分为三种判定方式。在判定过程中应注意边界问题。

1.直接连通

即两个方块间可以用一条直线连接。处在同一行或者同一列且中间没有其他方块。

2.单拐点连通

即两个方块间可以用两条线段连接。以两个方块为顶点得到一个矩形，矩形的另外两个顶点中任意一个空白顶点所在的位置与这两个方块都可以直接连通，则可连接。

3.双拐点连通

即两个方块间可以用三条线段连接。搜索其中一个方块周围四个方向空白的点，若任意一个点能与另一个方块单拐点连通，则能够连通。

4.寻找连通点

方案一

暴力求解，不推荐

四重循环，遍历每一个方块，起点遍历一次，终点遍历一次，判断起点和终点的连通性。

方案二

广搜

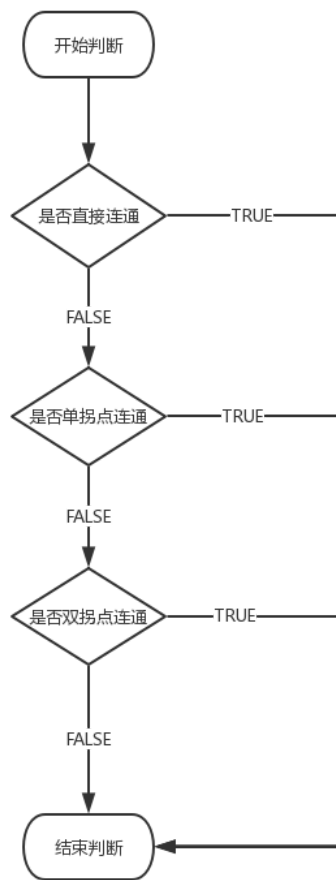
对每个方块进行广搜判断，两种判断方法

1. 方向改变不大于三次内可以找到的终点即可以连通，连通类型根据方向改变次数可以得出，拐点和方向改变次数每一次搜索都需要记录。（稍微麻烦，但速度快一点）
2. 对找到的终点进行连通性判断，不需要额外记录。（简单，相对慢一点，推荐）

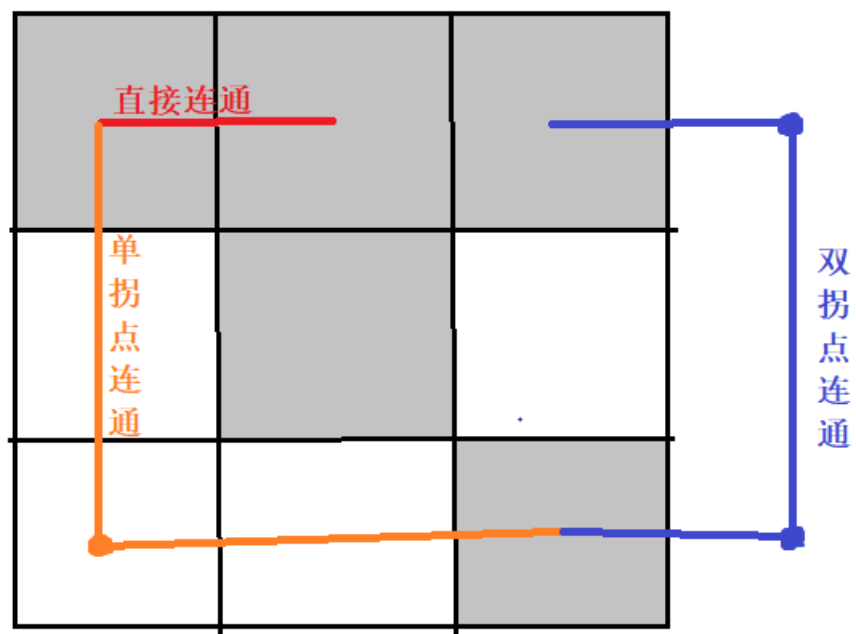
5.数据随机

将所有决定好的方块ID放入一个容器中进行混排，每次随机从容器中抛出一个ID设置在Map上，容器内的ID不能重复抛出。

判断流程图



连通示意图



类的封装

LinkType

枚举类型，表示四种方块的连通方式（包括一种无法连通）。

- STRAIGHT_LINK
- SINGLECORNER_LINK
- DOUBLECORNER_LINK
- NO_LINK

GameMap

保存方块数据的类。负责判断连通，提供数据和信号给界面程序。构造时应指定地图大小和方块ID最大值。

接口：

- 判断是否连通的方法，并记录所有拐点，返回[LinkResult](#)。
- 地图数据的getter方法，返回方块ID。
- 清空指定两个点的方法。
- 判断并清空两个点的方法，返回布尔类型。
- 重新随机地图的方法。
- 自动寻找两个可相消的点。成功返回一个[LinkResult](#)，失败抛出RuntimeException（或自定义一个Exception）。

LinkResult

自定义封装类，用于返回五个参数（初始化为null），为每个参数设置一个getter/setter。

- firstPoint
- secondPoint
- firstCorner
- secondCorner
- linkType