

# Java 作业说明

2019 年 5 月 9 日

## 1 作业综述

本次作业实现了一个含窗口界面的小游戏，游戏的目的是将一个随机产生的、有明暗两种状态的 6\*6 方阵全部点亮。每一次操作，玩家可以点击方阵中的一个按钮，将其和上下左右四个方格的状态改变（由点亮改成熄灭，或由熄灭改成点亮）。右上角有 help 键，点击会出现有操作提示的对话框；同时还有步数的计数器，会随着玩家的操作不断增加，直到点亮所有的方格，此时会有文字提示玩家通关。完成一局或厌倦当前局面时，玩家都可以点击左上角的 restart 键重新生成一局游戏；同时，游戏还提供一首背景音乐（《渔舟唱晚》），可以通过点击左上角的 music 键播放。再按一次 music 键可以终止音乐。

## 2 实现概述

本次作业的项目是基于 eclipse 自带的 windowbuilder 构建的，包含一个 lightup 类（代码本身）和一个 bgm.wav 文件作为游戏的背景音乐。

lightup 类主要是通过设计模式完成，内部逻辑通过几个辅助的函数实现。lightup 类使用 gridBagLayout 布局，包含 7 行 6 列。第一行从左到右分别是重置按钮、音乐按钮、提示 JLabel（占两列宽）、帮助按钮和计数器。为美观起见，三个按钮都添加了系统默认 classpath 中自带的 icon。第二行至第七行均匀排列 36 个按钮，按钮的亮和灭通过 [/com/sun/javafx/scene/control/skin/caspian/images/vk-light.png](#) 和 [/com/sun/javafx/scene/control/skin/caspian/images/vk-dark.png](#)

两个 icon 的转换实现。按钮通过设置边界显示隐去了边缘，以提升视觉效果。

为了记录按钮当前的亮灭状态，内部储存了 JButton 数组和对应 tag 数组，以便于修改对应按钮，同时通过 initialize, victory, press, update 和 check 等函数实现内部逻辑。

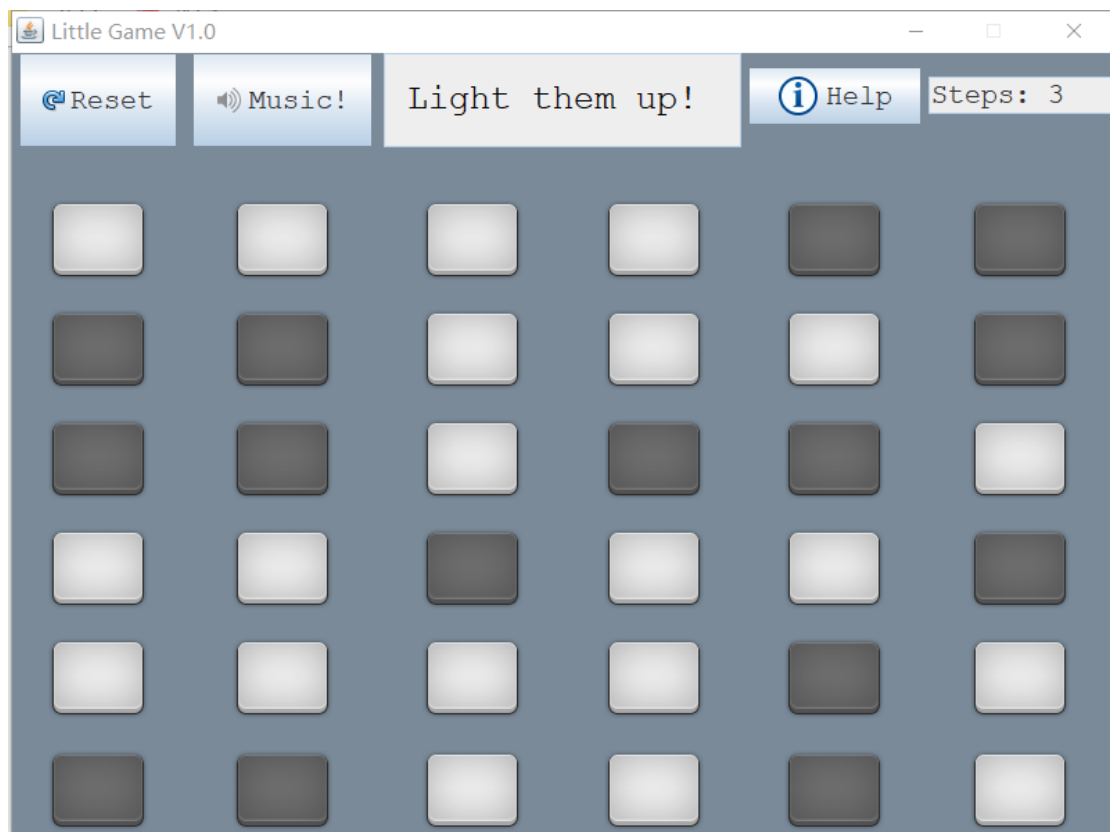
音乐播放用 audioclip 实现。按下按钮后音乐会一直循环播放

(2148483647 次) 直至再次按下按钮。第三次按下按钮后，音乐会重新从头播放。音乐的播放状态用 music\_on 标记记录。可能需要把音乐和代码放在同一个 src 文件夹下以正常播放音乐。

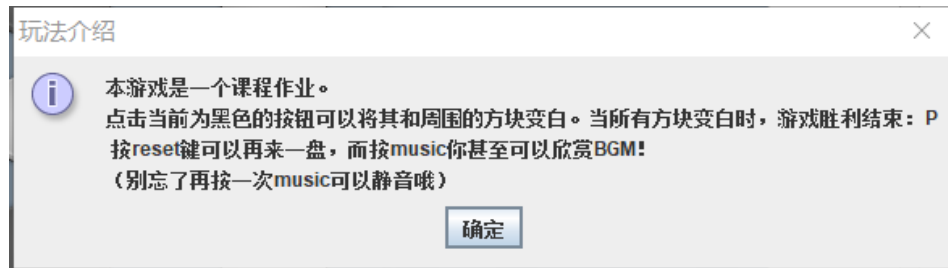
程序运行时不需要输入，直接点击运行即出现界面。

### 3 运行截图

游戏正常运行时的截图：



游戏帮助：



通关画面（注：修改内部逻辑得到的结果，只用于显示通关提示）

