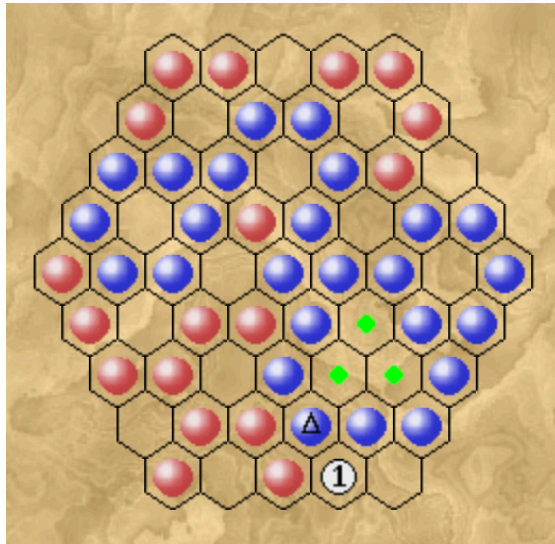


# Java程序设计大作业

## 作业简介

本次大作业的目标是实现衔环棋(Sabin Rains)，同学可以参考<https://www.di.fc.ul.pt/~jpn/gv/sabinrains.htm>。



## 游戏规则

Sabin Rains 是在一个5x5六边形棋盘上进行的。

- 六边形环：指棋盘上任何一个格子周围的六个格子组成的集合。六边形环中也可以少于六个格子，这种情况在棋盘边缘发生。
- 落子：
  - 红蓝双方交替，红方先手。每一轮，每个玩家在一个空格上放置一个棋子。
  - 棋子不能放在没有相邻空格的格子上。
  - 放置棋子后，新的棋子最多会出现在6个六边形环上，这些六边形环分别以该棋子相邻的格子为中心。
    - 在每个六边形环上，它会翻转被自己和一个友方棋子夹在该六边形环上的所有敌方棋子（触发翻转时，两个友方棋子之间必须全部是敌方棋子，而非空格）。
    - 如果六边形环上有五个敌方棋子时，落下的棋子会翻转所有五个。
    - 翻转规则独立作用在新棋子所处的6个六边形环上，也就是说不考虑某个六边形环翻转后对于其他六边形环的连锁效应。
- 目标：当棋盘上没有可落子的格子时，游戏结束，拥有更多棋子的玩家获胜。

## 例子

- 在最上方的图片中，假设目前是红方回合，并且红方在格子[1]处落子。格子[1]一共处在4个六边形环上，对应4个相邻格子。其中，只有在以[1]左侧红方棋子为中心的六边形环上触发了翻转规则，蓝方棋子△被翻转为红方。
- 轮到蓝方回合，此时棋盘上可落子的区域只有包含绿点的3个格子。蓝方在任意一处落子，不会触发翻转规则。
- 轮到红方回合，红方可以在绿点格子中的剩下2个落子。此时红方总能选择一处落子，翻转六边形环上的五个蓝方棋子，从而获胜。

# 需求描述

本节描述待实现的功能需求和评分细则。对于没有明确描述的功能或实现效果，选择任意方式实现均可。大作业必做内容总分为100分，另外有选做内容可以附加10分。

## 基本需求(95")

基本需求是作业中的必选部分。

### 游戏界面(35")

1. 绘制棋盘(12")：5x5六边形棋盘(4")，棋盘每格为正六边形(4")。棋盘背景颜色设置为白色以外的一种颜色，也不能与棋子的红色和蓝色重合，否则会覆盖棋子(4")。
2. 绘制棋子(12")：着色的圆形棋子。
  - 棋子为圆形(3")。
  - 棋子有两种颜色，红色和蓝色(3")。
  - 棋子处在格子正中央(3")。
  - 棋子不能与格子边线重叠(3")。
3. 棋盘提示(5")：棋盘不同区域有相应提示。
  - 可落子标识：可落子的格子中心用绿点标识(1")。
  - 待落子标识：鼠标所在的格子，如果确为可落子的格子，边框用不同于黑色的颜色高亮(1")。
  - 翻转标识：可以翻转的棋子所在格子的边框用不同于黑色的颜色和不同于待落子格子的颜色高亮(1")。
  - 锁定标识：锁定区域所在的格子染为灰色(1")，但是格子上的棋子颜色不受影响(1")。锁定区域指的是距离任何可落子的格子2格以外（不包含2格）的格子，这些格子上既无法落子，已有棋子的颜色也不可能再翻转。
4. 绘制游戏结果(6")：游戏结束时显示游戏结果。弹窗显示游戏结果(3")，只需要显示胜负结果即可(3")，例如显示内容为"WINNER: RED"或"WINNER: BLUE"。

### 游戏模型(45")

游戏模型要求精准实现。

1. 初始化(2")：按游戏规则初始化一个空棋盘。
2. 落子(16")：
  - 鼠标移动到待落子的空格，然后点击空格所在位置，根据游戏规则完成落子（不要求实现动画，一步到位即可）(12")。
  - 若落子位置根据游戏规则不合法，则不应该执行落子(4")。
3. 棋盘提示(6")：
  - 可落子标识和锁定标识始终存在，并且在落子瞬间进行更新(2")。
  - 鼠标移动到待落子格子上，立刻出现待落子标识。点击格子落子瞬间，待落子标识立即消失(2")。
  - 鼠标移动到待落子格子上，立刻出现待翻转标识。点击格子落子瞬间，翻转标识立即消失(2")。
4. 棋子翻转(4")：根据规则，落子瞬间，某些棋子颜色翻转（不要求实现动画，一步到位即可）。
5. 交替落子(2")：红棋先行，交替落子。轮到红棋时落子为红旗，轮到蓝棋时落子为蓝棋。
6. 胜负判定(10")：根据游戏规则，一旦某个落子后，触发游戏结束，则双方不能再落子(2")。然后进行胜负判断，并弹窗显示相应胜负结果(8")。
7. 计时(5")：需要在界面上绘制一个倒计时的计时器，以秒为单位倒计时(1")。红蓝双方必须在规定的30秒时间

内落子，一方落子瞬间，另一方立刻开始计时(2")。时间耗尽则随机在可落子区域内落子(2")。

## 性能需求(15")

程序运行有一些基本的性能需求，若不满足这些需求，则根据实际性能效果扣分。

1. 流畅性(7")：保证系统运行流畅、不卡顿、操作没有明显延迟，例如落子瞬间相应的棋盘提示发生响应，点击格子的瞬间完成落子。保证用户界面不发生闪烁。
2. 稳定性(8")：系统稳定运行，不崩溃，不卡死。测试时每发现一个问题扣2分，最多不超过8分。

## 扩展需求(10")

以下是可选的扩展需求，为加分项。请有兴趣的同学**选择一项**完成，拓展需求的分数直接加入大作业总分，最高为110分。

1. Robot(10")：蓝方改为程序自动落子，需要设计并实现一个落子策略，目的为尽可能获胜。分数判定依据策略效果和设计思路。需要在实验报告中阐明设计方法，并设计实验证明策略的效果。
2. 其他同学自主设计的拓展功能，请先联系助教确认。

## 实验报告(5")

实验报告为必选，包括以下内容：

1. （可选）扩展需求对应的设计方法和实验效果。
2. 使用类图或者自然语言描述项目的架构与设计(5")。
3. （可选）其他任何希望助教了解的内容，例如说明作业完成过程中参考的资料或个人。

## 提交格式

提交所有源代码，实验报告和一个可以直接运行的 jar 包（位置在根目录，缺少 jar 包大作业总分扣除10分），并打包为一个zip文件。

```
1  └─2000123456_张三.zip
2      └─ report.pdf //实验报告
3      └─  ***.jar    //可执行文件
4      └─  src        //源代码目录
```

## 注意事项

- 编程作业为个人作业，请独立完成。如果提交的代码参考了资料或个人，请在report中标注来源，否则视为抄袭，记0分。
- 使用Java语言，不限制使用的模块和库。但如果使用的不是Java内建的库，需要在jar包中一并打包。
- 如果有些功能实现复杂，例如棋盘提示，建议**优先保证整个游戏可以正常运行**，以便其他功能可以正常测试。