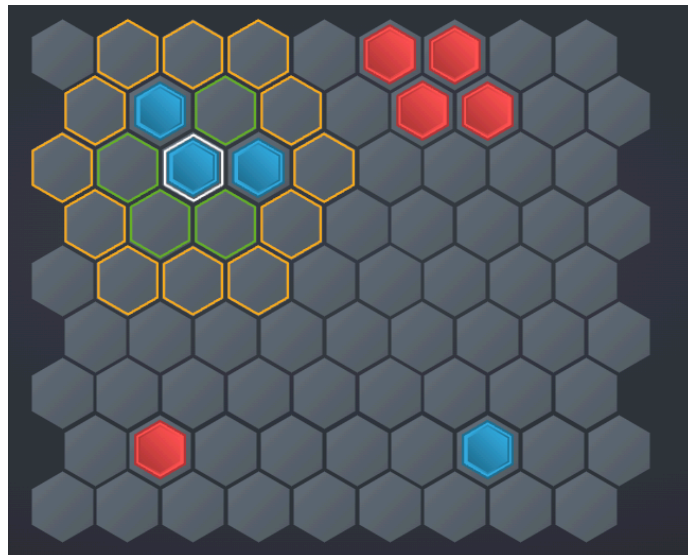


Java程序设计大作业

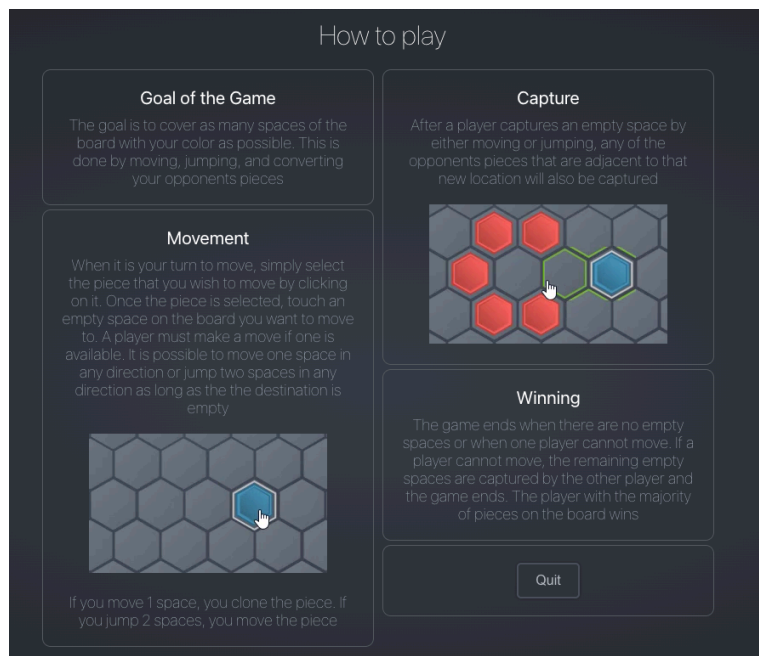
作业简介

本次大作业的目标是实现一个棋类游戏 Hex，建议同学参考<https://playhex.online>。

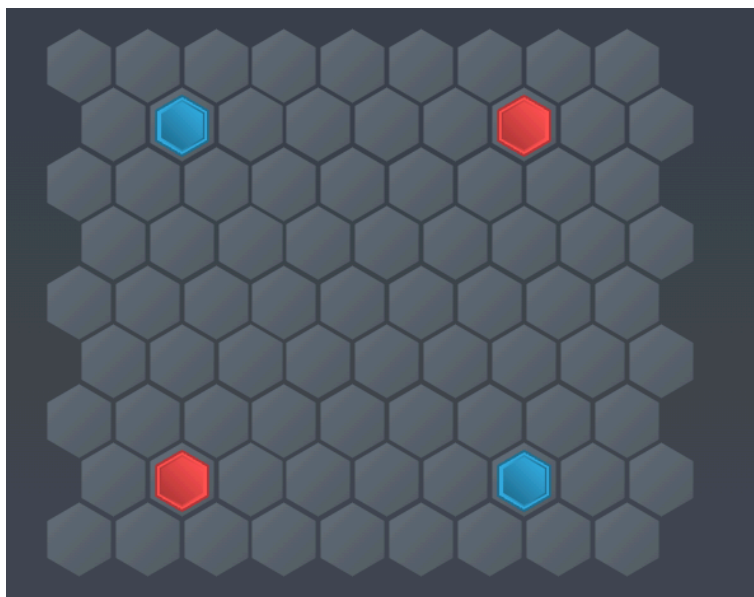


游戏规则

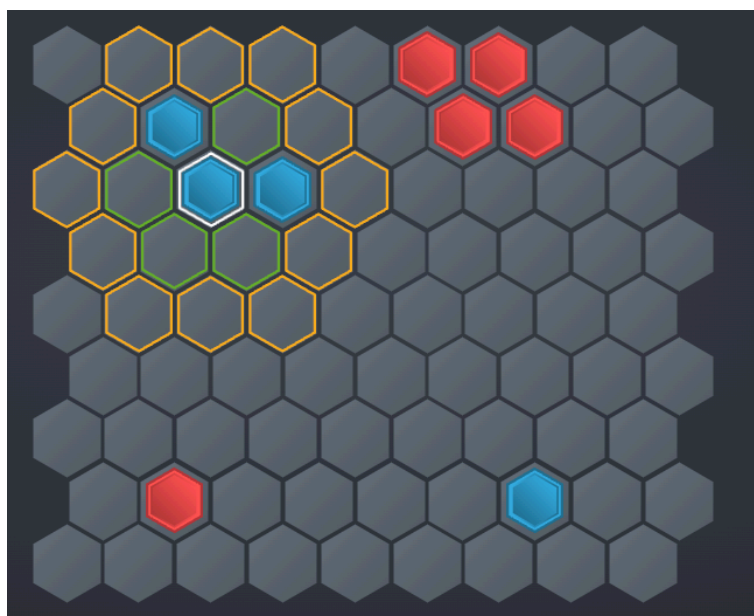
建议同学参考网页<https://playhex.online>的详细游戏规则说明，本文档对规则加以解释。



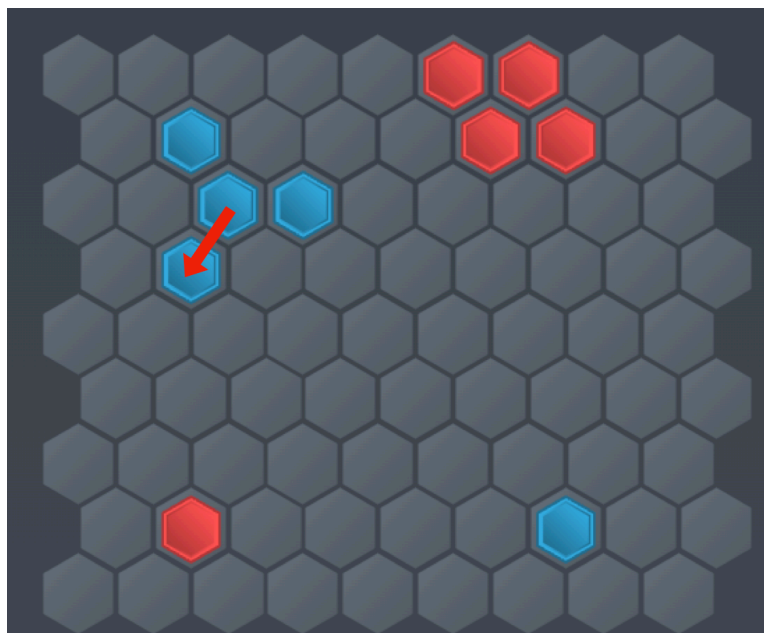
1. **棋盘：**棋盘形状 9×9 ，棋盘每格为正六边形。游戏共红蓝两方，分别执红棋和蓝棋。初始状态下，棋盘上有2颗蓝棋和2颗红棋，位置如下图所示。规定红棋先行。



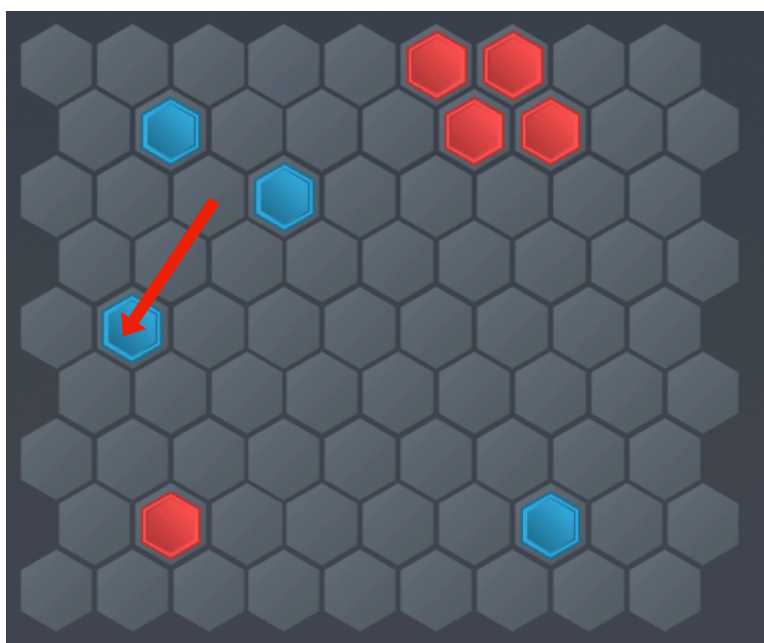
2. **移动**：红蓝两方轮流移动，红棋先行。每次移动时，玩家可选择己方的某颗棋子，向任意方向移动1-2格（落子处必须为棋盘上的空格）。例如下图为蓝方移动，白色高亮区域为选中的棋子，绿色高亮区域为移动1格时可能的落子点，黄色高亮区域为移动2格时可能的落子点。



若玩家选择移动1格（绿色高亮区域），则在目标网格落子后原来的棋子保留，即棋子个数增加一个，如下图。

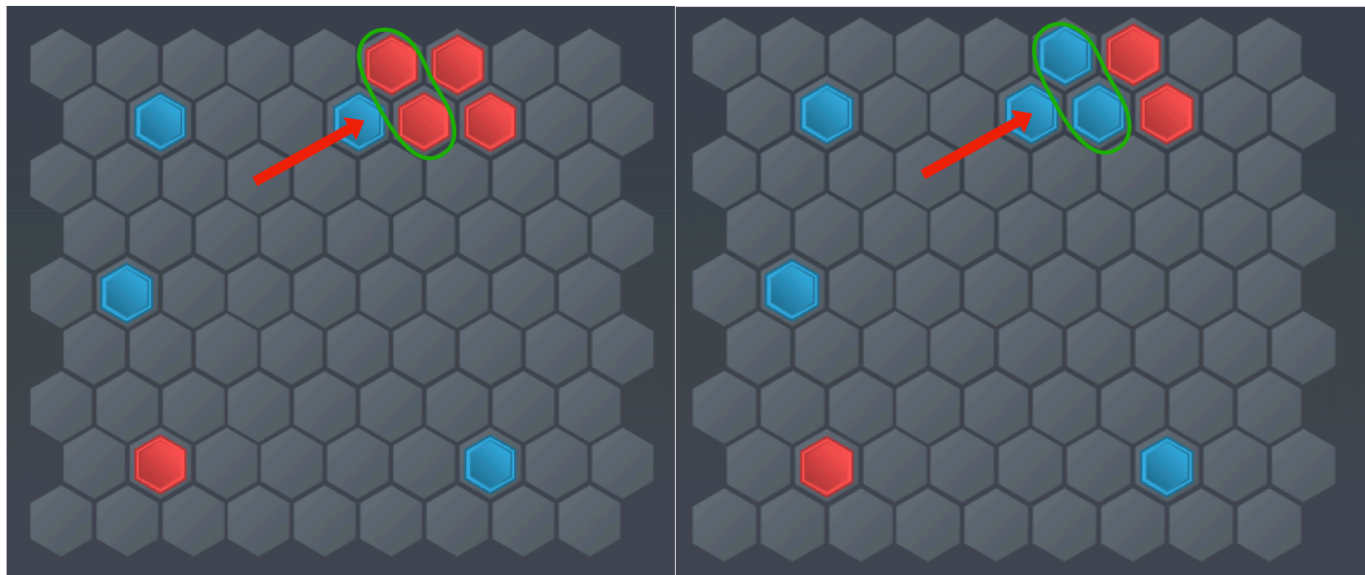


若玩家选择移动2格（黄色高亮区域），则在目标网格落子后原来的棋子消失，即棋子个数保持不变，如下图。



每次轮到一方时，必须选择一颗棋子进行移动，不能放弃移动或移动两颗棋子。

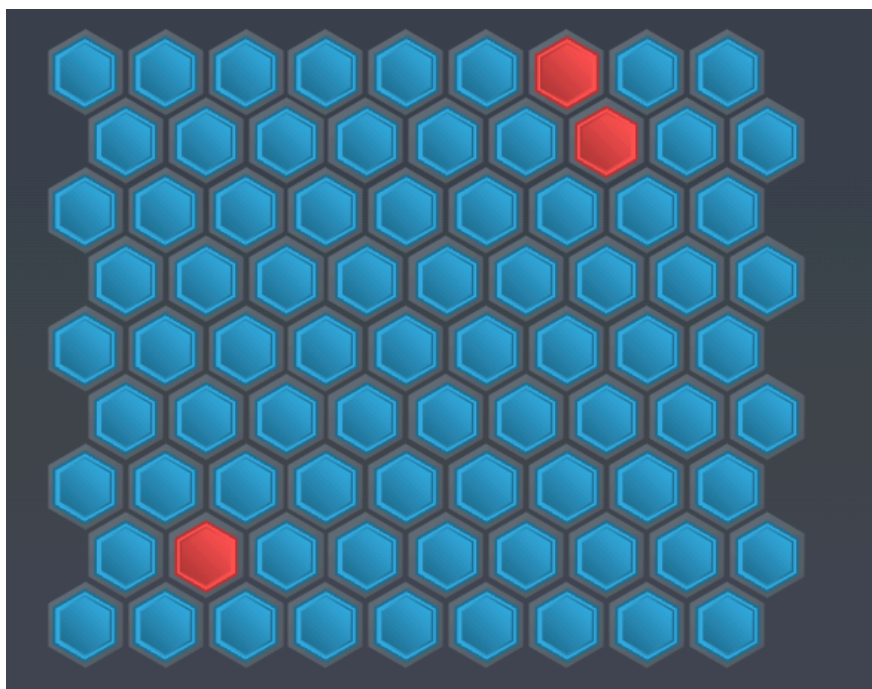
3. **占领：**一方移动时，若落子点相邻网格（1格相邻）内有另一方的棋子，这些棋子全部被俘获，即更换颜色，如下图。



4. 胜负判定：轮到某一方移动时，若其无法移动，则游戏结束。无法移动的情况可能有：

- 棋盘已满；
- 棋盘未满，但是某方棋子全部被俘获；
- 棋盘未满，但是某方棋子全部被“严密包围”，即其所有棋子的相邻两格内没有空格。

某方无法移动时，游戏结束，并在棋盘的剩余空格上填满另一方的棋子。完成填充后，棋盘上棋子多的一方获胜。由于网格个数为奇数，因此不存在平局的情况。例如下图中，蓝方获胜。



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20分。

基本需求(90")

基本需求是作业中的必选部分。

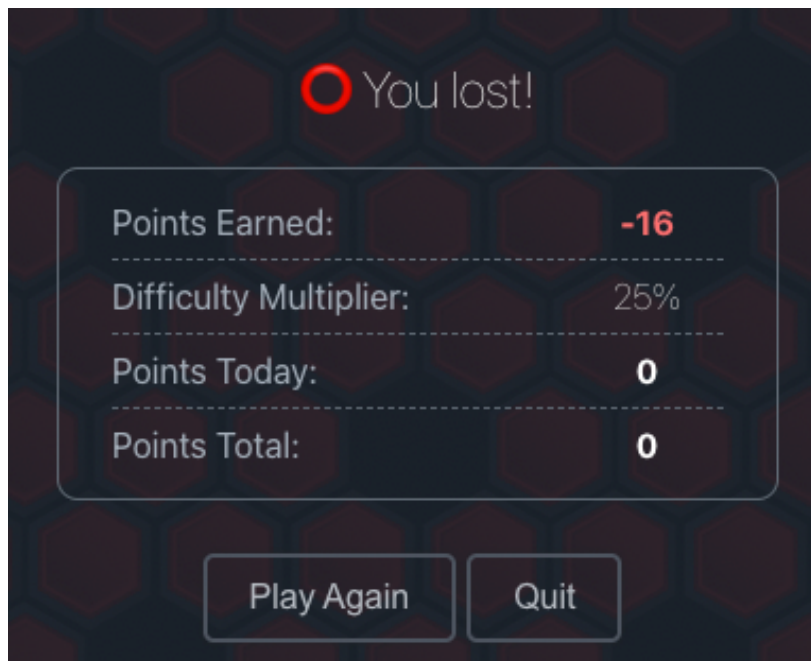
游戏界面(35")

游戏界面的设计较为灵活，对于游戏界面的某一项特征，同学可以选择不同完成度的实现效果，相对应给予不同评分。

1. 绘制棋盘(12")： 9×9 棋盘，棋盘每格为正六边形。棋盘和背景颜色不要求。
 - (12")棋盘和网格形状、相对位置正确，网格间有分界线（可实现为空隙或着色的线条）。
 - (10")棋盘和网格形状、相对位置正确，网格间没有分界线。
 - (6")棋盘或网格形状、相对位置不正确，但绘制了一些基本的六边形格子。
2. 绘制棋子(12")：着色的六边形棋子。
 - (12")棋子颜色正确（颜色根据RGB选择红色和蓝色即可）。棋子的形状为六边形，棋子的大小略小于棋盘网格大小，棋子位于网格的中心。
 - (10")棋子颜色正确（颜色根据RGB选择红色和蓝色即可）。棋子的形状为六边形，棋子的大小等于或超过了棋盘网格，或者棋子没有位于网格的中心。
 - (6")棋子颜色和棋子形状只有一个正确。
3. 棋子移动提示(5")：待移动的棋子周围有高亮提示。
 - (5")待移动的棋子，周围一格和周围两格有三种不同的高亮（可以按照示例实现为网格边界的染色，也可以选择其他方式，在棋盘网格上绘制一些差异化的特征即可，例如网格中心绘制不同颜色的圆点）。
 - (4")待移动的棋子，周围一格和周围两格使用同一种高亮。



4. 绘制游戏结果(6")：游戏结束时显示游戏结果。
 - (6")游戏结束时，弹窗显示游戏结果，只需要显示胜负结果即可，例如显示内容为"WINNER: RED"或"WINNER: BLUE"。
 - (5")游戏结束时，在原窗口界面上显示游戏结果。



游戏模型(40")

游戏模型要求精准实现。对于游戏执行逻辑的某一项功能，在单项分的基础上，测试时出现一处错误扣2分，最多扣分不超过单项分值。

1. 初始化(2")：按游戏规则初始化棋盘和四个棋子。
2. 棋子移动(16")：点击待移动棋子所在位置，然后点击空网格所在位置，根据游戏规则完成棋子移动（不要求实现动画，一步到位即可）。若落子位置根据游戏规则不合法，或点击了对方的棋子，则不应该执行棋子移动。
3. 棋子移动提示(6")：点击待移动棋子所在位置，在该棋子周围出现棋子移动提示效果，然后点击空网格所在位置完成移动后，相应提示效果消失。棋子移动提示应当是正确的，因此不合法的落子位置和棋盘外不应出现高亮效果。若点击了待移动棋子，然后又点击了另一个待移动棋子，则在另一个棋子周围出现提示效果。
4. 棋子俘获(4")：根据规则，棋子移动后，可能发生棋子俘获，此时被俘获的棋子需要发生变色（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。轮到红棋时只能移动红棋，轮到蓝棋时只能移动蓝棋。
6. 胜负判定(10")：根据游戏规则，一旦游戏结束，双方不能再移动棋子。根据游戏规则，需要在棋盘的剩余空格上填满棋子（不要求实现动画，一步到位即可）。然后进行胜负判断，并显示相应胜负结果。

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。

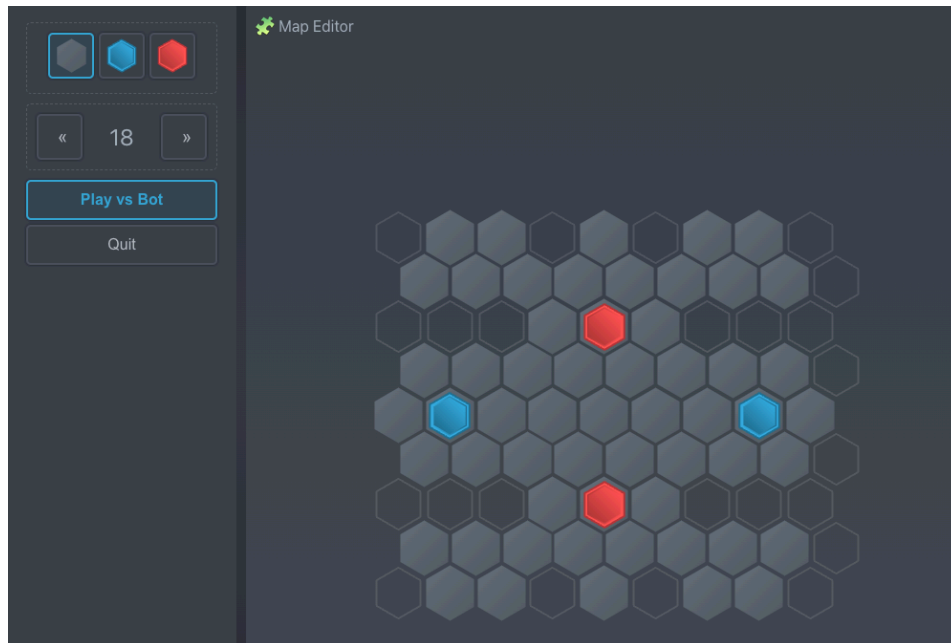
1. 流畅性(7")：保证系统运行流畅、不卡顿、操作没有明显延迟，例如点击待移动棋子的瞬间出现棋子移动提示，点击落子位置的瞬间完成落子。保证用户界面不发生闪烁。
2. 稳定性(8")：系统稳定运行，不崩溃，不卡死。测试时每发现一个问题扣2分，最多不超过8分。

扩展需求(20")

以下是可选的扩展需求，为加分项。有兴趣的同学请选择以下一项完成，拓展需求的分数直接加入大作业总分。完成多项扩展需求则计得分最高的一项。

1. 计时模式(5")：红蓝双方必须在规定的30秒时间内落子，一方落子后另一方立刻开始计时。时间耗尽则判定游戏结束，超时者负。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。

2. 动画(10"): 为棋子移动添加动画。参考<https://playhex.online>的动画效果。动画应当是平滑的, 不发生明显卡顿。
 - (5")棋子移动时, 消失的棋子有由大变小的渐变过程, 新出现的棋子有由小变大的渐变过程。
 - (2")棋子俘获时, 原先颜色的棋子有由大变小的渐变过程, 新出现颜色的棋子有由小变大的渐变过程。或者实现颜色的渐变过程。
 - (3")游戏结束时, 在棋盘的剩余空格填充棋子, 棋子逐格填充而不是同时填充, 填充过程有由小变大的渐变过程。
3. 定制棋盘(10"): 在游戏开始前, 可以通过用户界面定制不规则的棋盘形状。如下图, 实现效果参考<https://playhex.online>中的Map Editor。棋盘的网格形状和初始的四个棋子的位置均可以定制。



4. Robot(20"): 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。例如, 若自动程序对战<https://playhex.online>中Easy模式的Robot, 胜率超过90%, 则直接判定20分。
5. 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(10")

实验报告为必选, 包括以下内容:

1. 描述游戏操作方式(5"):
 - 列举实现的基础功能和加分项, 目的是使程序测试者可以发现并测试实现的每一个功能, 在实验报告中使用时序运行截图展示功能。
 - 对于较复杂的功能、亮点或加分项, 描述设计与实现方法, 并且引用相关模块在源代码中的位置 (文件名和行数)。对于加分项的评判同时考虑实现效果和实现方式, 因此这部分对于加分项的评判比较重要。
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语言和Java内建的包。
- 如果有些功能实现复杂，例如棋子移动提示，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。