

1 文件结构

本 lab 由 5 个文件构成，Lab3.java 中存放初始化、游戏主体及结束语句；Boarddata.java 中存放棋盘数据类相关方法；Board.java 中存放单一棋盘上的操作；Color.java 中存放棋盘用枚举类型；Player.java 中存放用户类和相关方法。

2 文件内容说明

2.1 Lab3.java

Line:8~Line:17 初始化游戏信息，输入双方玩家名、清屏打印空地图。

Line:20~Line:24 游戏主体循环，包含游戏逻辑、清屏、打印地图。

Line:27~Line:29 游戏结束输入缓冲区，打印结束语。

2.2 Boarddata.java

游戏棋盘相关方法。

2.2.1 存放数据

boardNum 棋盘数量。

currentBoard 当前棋盘编号（从 1 开始）。

board 棋盘数据。

2.2.2 构造方法

有参数构造方法中，参数输入棋盘高、宽，棋盘数量，黑方姓名，白方姓名。给棋盘数量赋值，当前棋盘编号赋初值：1。空棋盘用循环创建。

2.2.3 boardGame

游戏主体方法，通过 input 参数接收输入缓冲区。ifCorrect 用于记录用户输入的合法性。for 循环判断当前棋盘落子玩家并打印提示信息。while 循环接收玩家输入内容，当输入长度为 1 时，判断输入值是否合法，若合法则切换到指定棋盘，反之则输出错误提醒，重新输入。当输入长度为 2 时，进入下棋逻辑。当输入长度不合法时，输出错误提醒，重新输入。

2.3 Board.java

2.3.1 存放数据

height 地图的高。

width 地图的宽。

emptyRemain 地图上剩余空位。

boardCount 当前棋盘编号

currentColor 当前玩家颜色。

player 玩家数据。

boardColor 枚举类型棋盘。

boardString 字符类型棋盘。

2.3.2 构造类型

有参数构造方法接收地图高、宽，当前棋盘编号，黑方姓名，白方姓名。逐一赋值后构造两个棋盘，并通过循环赋初值。

2.3.3 colorTochar

将枚举类型棋盘逐一转换为字符类型棋盘正确位置。

2.3.4 printBoard

通过循环打印当前棋盘，通过分支语句在第 4 行打印棋盘编号，第 5、6 行打印玩家姓名和当前下棋状态。

2.3.5 boardUpdate

游戏下子逻辑方法，通过 move 参数接收玩家输入。将输入的两位分别通过内码转换成棋盘上行列，然后通过分支判断输入是否合法（是否在棋盘上，是否该位置已有子），输入合法的情况下将枚举类型棋盘对应位置赋对应值，棋盘空格减一，并更新字符类型棋盘，翻转当前玩家颜色。

2.4 Color.java

枚举类型，三个枚举值对应黑子、白子和空状态。

2.4.1 toString

通过 switch 分支获得枚举类型对应的字符。

2.4.2 convert

翻转当前颜色，若为空则不变。

2.5 Player.java

2.5.1 存放数据

name 玩家姓名。

`color` 玩家颜色。

3 运行截图

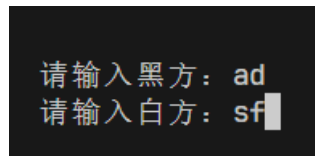


图 1: 玩家姓名输入

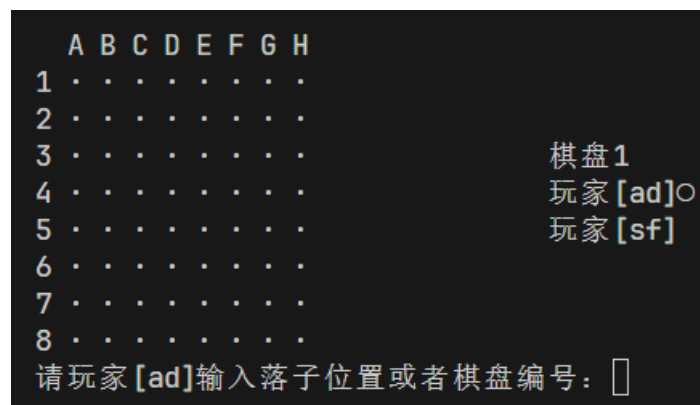


图 2: 空白棋盘 1



图 3: 空白棋盘 2

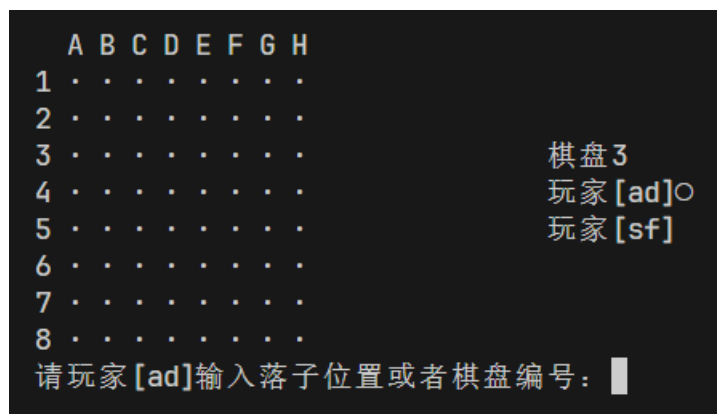


图 4: 空白棋盘 3

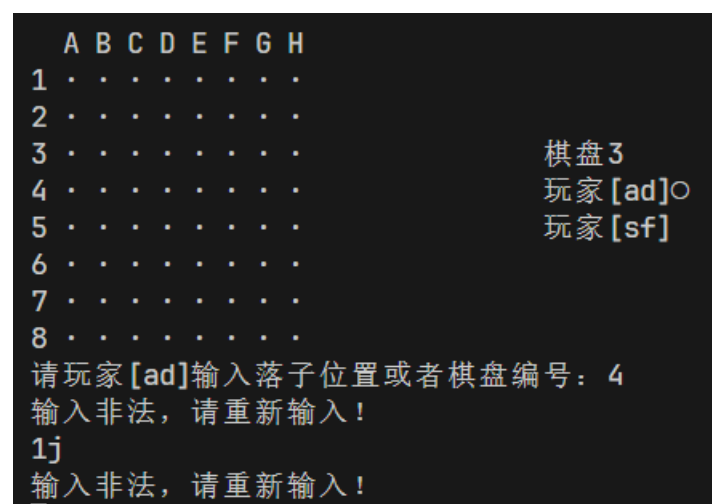


图 5: 非法输入检测



图 6: 结束界面