

## 1 文件结构

本 lab 由 4 个文件构成，Lab2.java 中存放初始化、游戏主体及结束语句；Mapdata.java 中存放地图类相关方法；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 Lab2.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 Mapdata.java

游戏棋盘相关方法。

#### 2.2.1 存放数据

**height** 棋盘的高。

**width** 棋盘的宽。

**emptyRemain** 棋盘剩余空格。

**currentColor** 当前下棋玩家的颜色。

**player** 玩家数据。

**mapColor** 枚举类型棋盘。

**mapString** 字符类型棋盘，包含行列索引。

#### 2.2.2 构造方法

有参数构造方法中，参数输入棋盘高、宽，黑方姓名，白方姓名。玩家数据用对象数组存放。空棋盘用循环创建，最后在字符类型棋盘的首行和首列通过内码添加行列索引。

### 2.2.3 colorTochar

将枚举类型棋盘逐一转换为字符类型棋盘正确位置。

### 2.2.4 printMap

通过循环打印游戏地图，通过分支语句在第 5、6 行打印玩家姓名和当前下棋状态。

### 2.2.5 mapUpdate

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，并跟新字符类型棋盘。

### 2.2.6 mapGame

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 循环判断当前落子玩家并打印提示信息。while 循环接收玩家输入内容，判断是否合法，合法情况进行下子逻辑。结束后将翻转当前玩家颜色。

## 2.3 Color.java

枚举类型，三个枚举值对应应有黑子、有白子和空状态。

### 2.3.1 toString

通过 switch 分支获得枚举类型对应的字符。

### 2.3.2 convert

翻转当前颜色，若为空则不变。

## 2.4 Player.java

### 2.4.1 存放数据

name 玩家姓名。

color 玩家颜色。

## 3 运行截图

```
请输入黑方：ad
请输入白方：sf
```

图 1: 玩家姓名输入

```
  A B C D E F G H
1 . . . . . . . .
2 . . . . . . . .
3 . . . . . . . .
4 . . . . . . . .
5 . . . . . . . .
6 . . . . . . . .
7 . . . . . . . .
8 . . . . . . . .
请玩家[ad]输入落子位置：

```

玩家[ad]○  
玩家[sf]

图 2: 空白棋盘

```
  A B C D E F G H
1 ○ ○ ○ ○ ○ . . .
2 ● ● ● ● ● . . .
3 ○ ○ ○ ○ ○ . . .
4 ● ● ● ● ● . . .
5 ○ ○ ○ ○ . . . .
6 ● ● ● ● . . . .
7 ○ ○ ○ ○ . . . .
8 ● ● ● ● . . . .
请玩家[ad]输入落子位置：
4e
输入非法，请重新输入！
5e
```

玩家[ad]○  
玩家[sf]

图 3: 非法输入检测

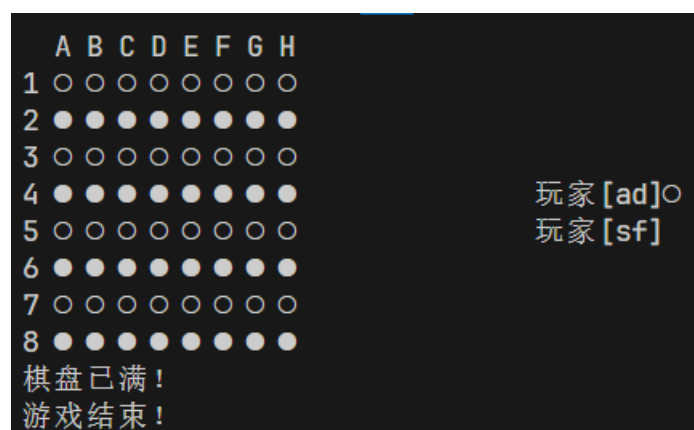


图 4: 结束界面