(1) 项目说明

本项目是一个简单的黑白棋游戏，支持两名玩家轮流在 8x8 的棋盘上落子。游戏规则基于经典黑白棋，但未实现复杂的翻转规则，如果棋盘已满，游戏结束。（本次 lab 无判断胜负）。项目使用 Java 编写，主要分为模型、视图和控制三个部分。

---

(2) 源代码文件及其主要功能

以下是项目中的主要文件及其功能：

文件名称	主要功能
App.java	程序入口，负责初始化游戏，创建玩家并启动游戏引擎。
GameEngine.java	游戏引擎，负责游戏逻辑，包括玩家轮流落子、判断游戏结束等。
GameView.java	负责显示游戏界面，包括棋盘、玩家信息和当前玩家提示。
InputUtils.java	负责处理用户输入，验证输入格式并将其转换为棋盘坐标。
Board.java	棋盘模型，负责初始化棋盘、存储棋子状态以及判断落子是否合法。
Piece.java	棋子枚举类，定义棋子类型（空、黑、白）及其符号表示。
Player.java	玩家类，存储玩家信息（名称、棋子类型）并提供相关方法。

---

(3) 关键代码及其设计思路

项目大体上采用了类似 MVC（模型-视图-控制器）的设计思路。

- 模型层： 包括棋盘（Board）、棋子（Piece）和玩家（Player）的定义，负责存储游戏状态。
- 视图层： 由 GameView 负责展示棋盘状态和提示信息。
- 控制层： 由 GameEngine 管理游戏流程、玩家回合以及调用输入工具和视图更新。

1. 输入校验与落子检查（InputUtils.java 中的 readValidInput 方法）

这个方法将输入解析和合法性检测（包括落子合法性）整合到了一起，既避免了重复代码，也使得游戏引擎能专注于游戏逻辑：

```
public static int[] readValidInput(Scanner scanner, Board board, Piece piece) {
    while (true) {
        String input = scanner.nextLine().trim();
        if (input.isEmpty()) continue; // 忽略空行
        try {
            int[] move = parseInput(input);
            int row = move[0];
            int col = move[1];
            if (!Board.canplacePiece(row, col, piece)) {
                System.out.println("This position cannot be placed. Please try again.");
                continue;
            }
            return move;
        } catch (IllegalArgumentException e) {
            System.out.println("Invalid input format. Please enter a number (1-8) followed by a letter (A-H).");
        }
    }
}
```

设计亮点：

集中式验证： 输入格式和落子可行性在一个方法中检查，避免在多个地方重复验证逻辑。

用户反馈： 错误信息明确，及时指导玩家输入正确的指令。

2. 玩家回合处理（GameEngine.java 中的 handleTurn 方法）

游戏引擎在处理每个回合时，调用上述输入工具获得一个合法落子，然后直接更新棋盘状态，使逻辑清晰易懂：

```
private void handleTurn(Player player) {
    GameView.printBoard(board, players[currentPlayerIndex], players[0], players[1]);
    int[] position = InputUtils.readValidInput(scanner, board, player.pieceType);
    int row = position[0];
    int col = position[1];
    Board.grid[row][col] = player.pieceType;
}
```

设计亮点：

职责分离： 游戏引擎只负责调用视图和更新状态，具体的输入验证交由 InputUtils 处理。

简化逻辑： 使用封装好的方法后，回合处理代码简单明了。

3. 跨平台控制台清屏（GameView.java 中的 clearConsole 方法）

为了改善用户体验，清屏方法根据操作系统选择不同的清屏策略，既兼顾 Windows 平台又适用于 Unix 系统（因為本人用 Mac）：

```
public static void clearConsole(){
```

```
    try {
        final String os = System.getProperty("os.name").toLowerCase();
        if (os.contains("win")) {
            new ProcessBuilder("cmd", "/c", "cls").inheritIO().start().waitFor();
        } else {
            System.out.print("\033[H\033[2J");
            System.out.flush();
            IntStream.range(0, 50).forEach(i -> System.out.println());
        }
    } catch (IOException | InterruptedException e) {
        System.err.println("Clear console failed: " + e.getMessage());
    }
}
```

设计亮点：

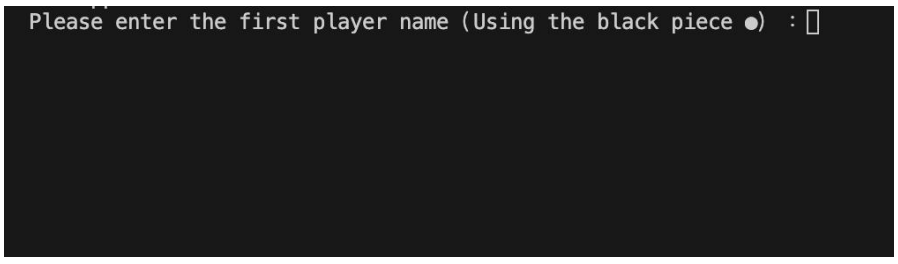
平台适应性： 通过检测操作系统名称，使用对应的清屏方式确保在不同平台下都能正常运行。

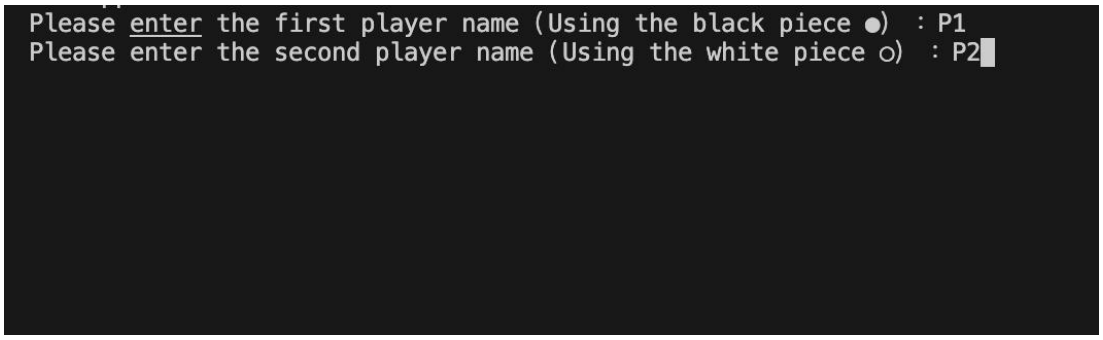
鲁棒性： 异常捕获确保即使清屏失败，程序依然能够继续运行。

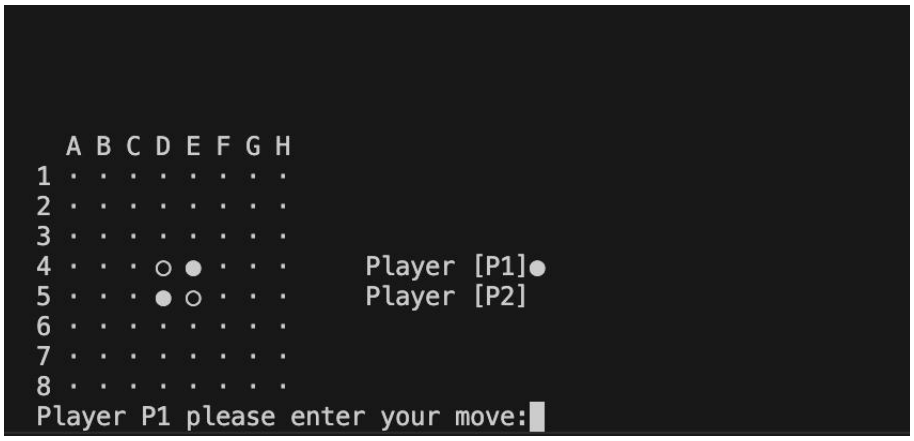
---

(4) 运行过程截图及说明

4.1 游戏启动界面

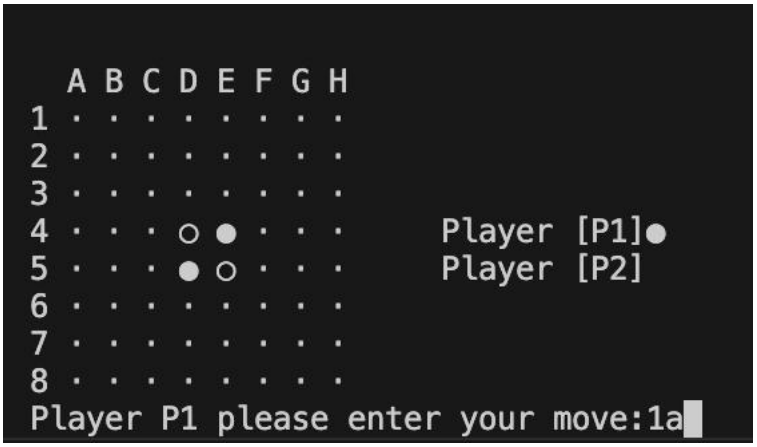


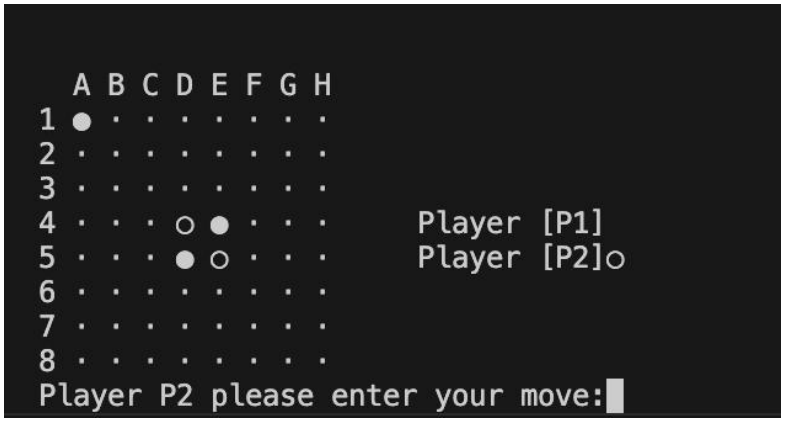




- 说明：程序启动后，提示玩家输入姓名。

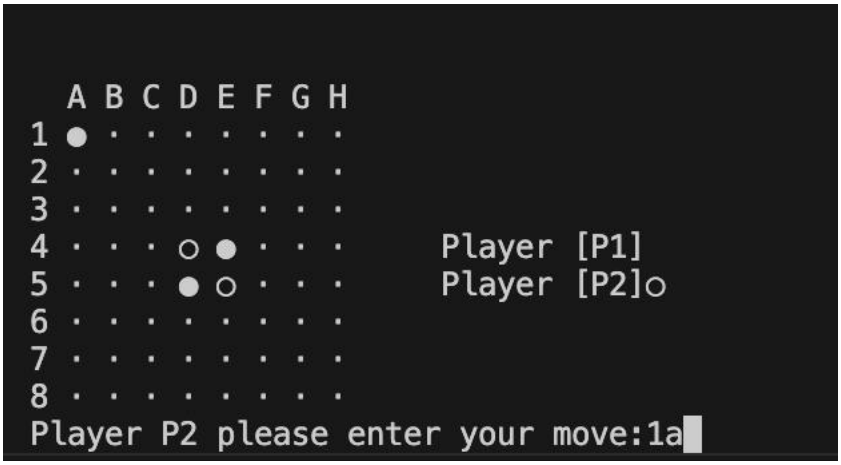
4.2 游戏进行界面

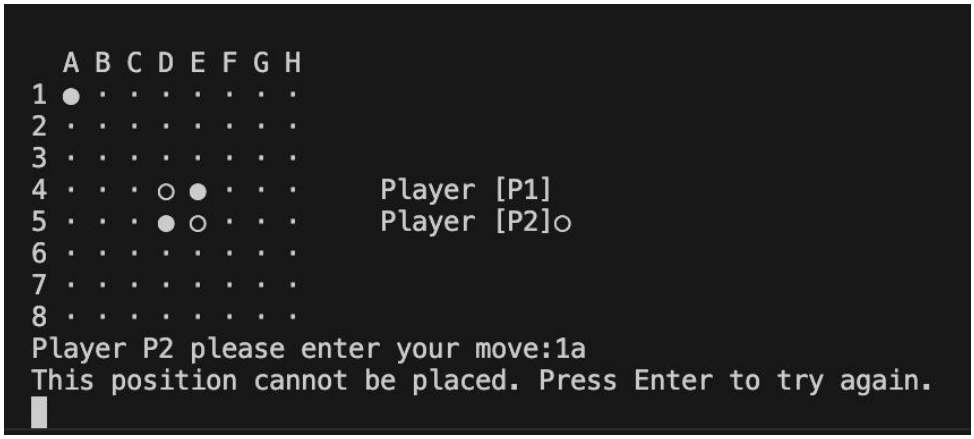




- 说明：显示棋盘、当前玩家信息及落子提示。玩家输入坐标后，程序更新棋盘并切换玩家。

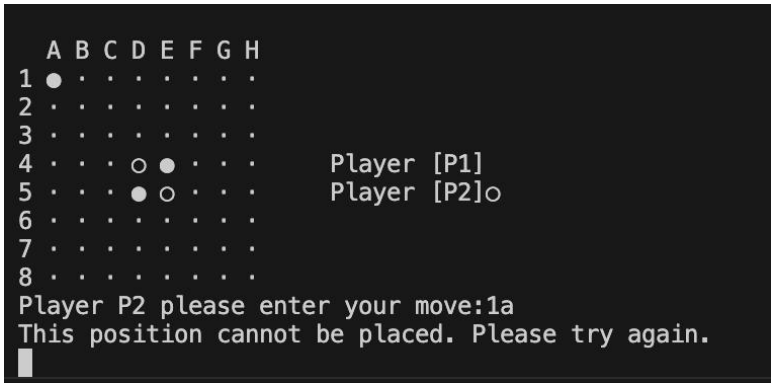
4.3 游戏结束界面



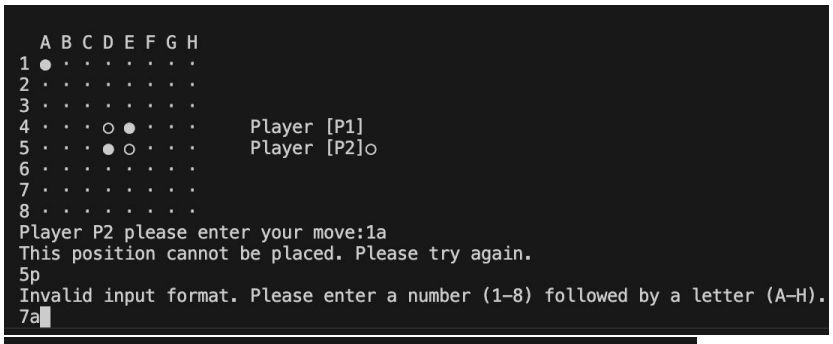


- 说明：当棋盘填满时，游戏结束，显示结束信息。

4.4 游戏输入异常


A B C D E F G H
1 ● · · · · · · ·
2 · · · · · · · ·
3 · · · · · · · ·
4 · · · ○ ● · · ·
5 · · · ● ○ · · ·
6 · · · · · · · ·
7 · · · · · · · ·
8 · · · · · · · ·
Player [P1]
Player [P2]○
Player P2 please enter your move:1a
This position cannot be placed. Please try again.


A B C D E F G H
1 ● · · · · · · ·
2 · · · · · · · ·
3 · · · · · · · ·
4 · · · ○ ● · · ·
5 · · · ● ○ · · ·
6 · · · · · · · ·
7 · · · · · · · ·
8 · · · · · · · ·
Player [P1]
Player [P2]○
Player P2 please enter your move:1a
This position cannot be placed. Please try again.
5p
Invalid input format. Please enter a number (1-8) followed by a letter (A-H).


A B C D E F G H
1 ● · · · · · · ·
2 · · · · · · · ·
3 · · · · · · · ·
4 · · · ○ ● · · ·
5 · · · ● ○ · · ·
6 · · · · · · · ·
7 · · · · · · · ·
8 · · · · · · · ·
Player [P1]
Player [P2]○
Player P2 please enter your move:1a
This position cannot be placed. Please try again.
5p
Invalid input format. Please enter a number (1-8) followed by a letter (A-H).
7a


A B C D E F G H
1 ● · · · · · · ·
2 · · · · · · · ·
3 · · · · · · · ·
4 · · · ○ ● · · ·
5 · · · ● ○ · · ·
6 · · · · · · · ·
7 ○ · · · · · · ·
8 · · · · · · · ·
Player [P1]●
Player [P2]
Player P1 please enter your move:

- **说明**：图一为已有棋在输入的位置，图二为错误输入格式，图三四为正确输入。

---

(5) 5. 总结

本项目实现了一个简单的黑白棋游戏，涵盖了输入处理、游戏逻辑和界面显示等核心功能。通过模块化设计，代码结构清晰，易于扩展和维护。未来可以添加更多功能，如翻转规则、计分系统等，进一步提升游戏体验。