实验报告：Reversi 和 Peace 游戏系统

执行 Java 程式

```
java -cp bin App
```

项目说明

本项目是一个基于 Java 的多游戏系统，实现了两种棋盘游戏：**Reversi（黑白棋）** 和 **Peace（简化版棋盘游戏）**。系统支持多个游戏同时进行，并提供以下功能：

1. **玩家交互**：支持玩家输入名称、选择游戏并进行操作。
2. **游戏逻辑处理**：
   - Reversi：实现棋子合法性检查、翻转逻辑和胜负判断。
   - Peace：简化的棋盘游戏，允许玩家在空位置放置棋子。
3. **棋盘显示**：动态更新棋盘状态，显示玩家分数和当前游戏状态。
4. **多游戏管理**：支持同时管理多个游戏，玩家可在不同游戏间切换。

该系统采用了模块化设计，代码结构清晰，便于扩展和维护。

源代码文件及其主要功能

源代码文件	功能	主要逻辑
App.java	程序的入口点，负责启动游戏。	创建 Scanner 对象，用于读取用户输入，调用 GameManager.start(scanner) 开始游戏。
GameManager.java	管理多个游戏的逻辑，包括游戏初始化、切换游戏、处理玩家输入等。	初始化玩家信息；创建并管理多个游戏（Peace 和 Reversi）；处理玩家输入（如移动、切换游戏、退出等）；调用 GameView 显示游戏状态。
GameEngine.java	定义游戏引擎的接口，为所有游戏提供统一的操作方法。	定义游戏基本操作，如放置棋子、检查游戏是否结束、获取棋盘状态等。
Reversi.java	实现 Reversi 游戏的逻辑。	使用方向数组检查棋子合法放置位置；翻转被夹住的棋子；计算棋盘上各棋子的数量。
Peace.java	实现 Peace 游戏的逻辑，简化版棋盘游戏。	允许玩家在空位置放置棋子；检查棋盘是否已满。
GameSetup.java	负责初始化玩家信息。	提示用户输入玩家名称；分配棋子类型（黑棋或白棋）。
GameView.java	负责游戏视觉化显示，包括棋盘打印、玩家信息显示等。	打印棋盘状态；显示当前玩家、分数和游戏状态；提供清屏功能。

源代码文件	功能	主要逻辑
InputUtils.java	处理用户输入，解析输入格式并进行验证。	支持多种输入格式（如棋盘坐标、切换游戏、退出等）；验证输入是否合法。
Board.java	表示棋盘的数据结构，提供棋盘操作方法。	初始化棋盘；设置和获取棋盘上的棋子；检查棋盘是否已满。
Piece.java	定义棋子的类型（如黑棋、白棋、空位等）。	使用枚举类型表示不同棋子；提供棋子的符号表示。
Player.java	表示玩家信息。	存储玩家名称和棋子类型；提供玩家信息的格式化输出。

关键代码及其设计思路

1. **Reversi 的合法性检查**

```
public boolean canPlacePiece(Piece piece) {
       // 清除棋盘上的 CANPLACE 标记
       for (int i = 0; i < Board.SIZE; i++) {
           for (int j = 0; j < Board.SIZE; j++) {
               if (board.getWhatPiece(i, j) == Piece.CANPLACE) {
                   board.setPiece(i, j, Piece.EMPTY);
               }
           }
       }

       boolean canPlace = false;
       for (int i = 0; i < Board.SIZE; i++) {
           for (int j = 0; j < Board.SIZE; j++) {
               // 检查当前位置是否为空
               if (board.getWhatPiece(i, j) != Piece.EMPTY) {
                   continue;
               }

               boolean isValidMove = false;

               // 遍历所有方向
               for (int k = 0; k < directions.length; k++) {
                   int x = directions[k][0];
                   int y = directions[k][1];
                   int steps = 0;

                   // 沿着方向检查
                   while (i + x >= 0 && i + x < Board.SIZE && j + y >= 0
&& j + y < Board.SIZE) {
                       if (board.getWhatPiece(i + x, j + y) == Piece.EMPTY
||
                               board.getWhatPiece(i + x, j + y) ==
Piece.CANPLACE) {
                           break;
```

```
                        }

                        if (board.getWhatPiece(i + x, j + y) == piece) {
                            if (steps > 0) { // 确保至少有一个棋子被夹住
                                isValidMove = true;
                            }
                            break;
                        }

                        steps++;
                        x += directions[k][0];
                        y += directions[k][1];
                    }
                }

                // 如果该位置合法，标记为 CANPLACE
                if (isValidMove) {
                    board.setPiece(i, j, Piece.CANPLACE);
                    canPlace = true;
                }
            }
        }

        return canPlace;
    }
```

设计思路

- 使用方向数组（`directions`）遍历每个可能的方向。
- 检查是否存在至少一个夹住对方棋子的情况。
- 如果合法，将该位置标记为 `CANPLACE`。

2. **Reversi 的棋子翻转逻辑**

```
public void placePiece(int col, int row, Piece piece) {
    passCounter = 0;
    // 确保当前位置为合法位置
    if (board.getWhatPiece(col, row) == Piece.CANPLACE) {
        board.setPiece(col, row, piece);

        for (int k = 0; k < directions.length; k++) {
            int dx = directions[k][0];
            int dy = directions[k][1];
            int x = dx;
            int y = dy;
            // 用于存储需要翻转的棋子位置
            java.util.ArrayList<int[]> toFlip = new java.util.ArrayList<>
();
            boolean validDirection = false;

            while (col + x >= 0 && col + x < Board.SIZE && row + y >= 0 &&
```

```
row + y < Board.SIZE) {
                Piece currentPiece = board.getWhatPiece(col + x, row + y);

                if (currentPiece == Piece.EMPTY || currentPiece ==
Piece.CANPLACE) {
                    break;
                } else if (currentPiece == piece) {
                    validDirection = true;
                    break;
                } else {
                    toFlip.add(new int[] { col + x, row + y });
                }

                x += dx;
                y += dy;
            }

            // 如果方向合法，翻转所有记录的棋子
            if (validDirection) {
                for (int[] pos : toFlip) {
                    board.setPiece(pos[0], pos[1], piece);
                }
            }
        }
    }
}
```

设计思路

- 遍历所有方向，记录需要翻转的棋子位置。
- 当找到合法方向时，翻转所有记录的棋子。

3. InputUtils 的输入解析与验证

```
public static int[] parseInput(String input) throws
IllegalArgumentException {
    // 移除不可打印字符并修剪输入
    input = input.replaceAll("[^\\p{Print}]", "").trim();

    if (input.equalsIgnoreCase("quit")) {
        return new int[] { 0, -2 }; // 用户选择退出游戏
    }

    if (input.equalsIgnoreCase("pass")) {
        return new int[] { 0, -3 }; // 用户选择跳过回合
    }

    if (input.equalsIgnoreCase("peace")) {
        return new int[] { 0, -4 }; // 用户选择开始 Peace 游戏
    }
```

```
    if (input.equalsIgnoreCase("reversi")) {
        return new int[] { 0, -5 }; // 用户选择开始 Reversi 游戏
    }

    // 检查是否为单个数字，表示棋盘索引
    if (input.matches("-?\\d+")) {
        int boardIndex = Integer.parseInt(input) - 1;
        if (boardIndex < 0 || boardIndex >=
GameManager.getNumberOfGames()) {
            throw new IllegalArgumentException("Invalid board index");
        }
        return new int[] { boardIndex, -1 }; // 返回棋盘索引
    }

    // 检查是否为合法的棋盘坐标
    Matcher matcher = INPUT_PATTERN.matcher(input);
    if (!matcher.find()) {
        throw new IllegalArgumentException("Invalid input format");
    }

    int row = Integer.parseInt(matcher.group(1)) - 1;
    int col = Character.toLowerCase(matcher.group(2).charAt(0)) - 'a';

    return new int[] { row, col }; // 返回行和列
}
```

设计思路

- **输入格式处理：** 移除不可打印字符，确保输入的清洁性。 支持多种输入命令（如 quit、pass、peace、reversi 等）。
- **棋盘索引检查：** 如果输入为数字，则将其解析为棋盘索引，并检查是否在有效范围内。
- **棋盘坐标解析：** 使用正则表达式匹配合法的棋盘坐标（如 1A 或 8H）。 将坐标转换为数组形式，方便后续处理。

4. InputUtils 的输入读取与验证

```
public static int[] readValidInput(Scanner scanner, GameEngine engine,
Piece piece) {
    while (true) {
        String input = scanner.nextLine().trim();
        if (input.isEmpty())
            continue; // 忽略空行
        try {
            int[] move = parseInput(input);
            if (move[1] <= -1) {
                return move; // 返回特殊命令（如退出、跳过回合等）
            }
            // 检查是否可以放置棋子
            engine.canPlacePiece(piece);
```

```
            if (engine.getClass().getSimpleName().equals("Reversi")) {
                if (engine.getBoard().getWhatPiece(move[0], move[1]) ==
Piece.CANPLACE) {
                    return move; // 合法位置
                } else {
                    System.out.println("This position cannot be placed.
Please try again.");
                    continue;
                }
            }
            if (engine.getClass().getSimpleName().equals("Peace")) {
                if (engine.getBoard().getWhatPiece(move[0], move[1]) ==
Piece.EMPTY) {
                    return move; // 合法位置
                } else {
                    System.out.println("This position cannot be placed.
Please try again.");
                    continue;
                }
            }
        } catch (IllegalArgumentException e) {
            System.out.printf(
                    "Invalid input format. Please enter a number (1-8)
followed by a letter (A-H) / a board number (1-%d) / new game (peace /
reversi) / quit the game (quit): ",
                    GameManager.getNumberOfGames());
        }
    }
}
```

设计思路

- **循环读取输入：** 使用 Scanner 循环读取用户输入，忽略空行。
- **解析输入：** 调用 parseInput 方法解析输入，处理特殊命令（如退出、跳过回合等）。
- **合法性检查:** 根据当前游戏类型（Reversi 或 Peace），检查输入位置是否可以放置棋子。 如果位置不合法，提示用户重新输入。
- **错误处理：** 捕获非法输入的异常，提示用户正确的输入格式。

5. GameView 的棋盘显示与游戏状态管理

```
public static void printBoard(GameEngine engine, Board board, Player
currentPlayer, Player blackPlayer,
        Player whitePlayer) {
    clearConsole();

    String blackplayerStr = null;
    String whiteplayerStr = null;
```

```
    String blackPlayerScore = null;
    String whitePlayerScore = null;

    if (engine instanceof Reversi) {
        whitePlayerScore =
Integer.toString(GameManager.getCurrentGame().getHowManyPieces(whitePlayer
.pieceType));
        blackPlayerScore =
Integer.toString(GameManager.getCurrentGame().getHowManyPieces(blackPlayer
.pieceType));
    }

    whiteplayerStr = String.format("Player [%s] ", whitePlayer.getName());
    blackplayerStr = String.format("Player [%s] ", blackPlayer.getName());

    int nameLengthDiff = Math.max(blackplayerStr.length(),
whiteplayerStr.length())
            - Math.min(blackplayerStr.length(),
                    whiteplayerStr.length());
    if (nameLengthDiff > 0) {
        if (blackplayerStr.length() > whiteplayerStr.length()) {
            whiteplayerStr = String.format("%s%s", whiteplayerStr, "
".repeat(nameLengthDiff));
        } else {
            blackplayerStr = String.format("%s%s", blackplayerStr, "
".repeat(nameLengthDiff));
        }
    }

    if (currentPlayer == whitePlayer) {
        whiteplayerStr = String.format("%s" + whitePlayer.getpieceType(),
whiteplayerStr);
    } else if (currentPlayer == blackPlayer) {
        blackplayerStr = String.format("%s" + blackPlayer.getpieceType(),
blackplayerStr);
    }

    System.out.println("  A B C D E F G H");
    for (int i = 0; i < Board.SIZE; i++) {
        String left = buildLeftSection(engine, board, i);
        String middle = buildMiddleSection(engine, blackPlayerScore,
whitePlayerScore, blackplayerStr,
                whiteplayerStr, i);
        String right = buildRightSection(i);

        System.out.printf("%-25s %-30s %-50s%n", left, middle, right);
    }

    // 判断游戏是否结束
    boolean allGamesOver = true;
    for (GameEngine game : GameManager.games) {
        if (!game.isGameOver()) {
            allGamesOver = false;
            break;
```

```
        }
    }

    if (engine instanceof Reversi) {
        if (engine.getBoard().isBoardFull() || (engine.isGameOver() &&
engine.getPassCounter() == 2)) {
            System.out.println("Game " + (engine.getGameID() + 1) + " is
over now.");
            if (GameManager.getCurrentGame() instanceof Reversi) {
                engine = GameManager.getCurrentGame();
                if (engine.getHowManyPieces(Piece.BLACK) >
engine.getHowManyPieces(Piece.WHITE)) {
                    System.out.println("Player [" + blackPlayer.getName()
+ "]  wins!");
                } else if (engine.getHowManyPieces(Piece.BLACK) <
engine.getHowManyPieces(Piece.WHITE)) {
                    System.out.println("Player [" + whitePlayer.getName()
+ "]  wins!");
                } else {
                    System.out.println("It's a tie!");
                }
            }
            if (allGamesOver) {
                System.out.println("All Games over! All the boards are
full.");
                System.out.print("Please enter a new game (peace /
reversi) / quit the game (quit) : ");
            } else {
                System.out.println(
                        "Please enter another board number to continue /
new game (peace / reversi) / quit the game (quit) : ");
            }
        } else {
            if (!engine.canPlacePiece(currentPlayer.pieceType) &&
engine.getPassCounter() < 2) {
                System.out.printf("Player " + currentPlayer.getName() +
                        ", you do not have place for your piece, please
enter pass / game number (1-%d) / new game (peace / reversi) / quit : ",
                        GameManager.getNumberOfGames());
                return;
            } else if (engine.getPassCounter() < 2) {
                System.out.printf("Player " + currentPlayer.getName() +
                        ", please enter your move (1-8,a-h) / game number
(1-%d) / new game (peace / reversi) / quit the game (quit) : ",
                        GameManager.getNumberOfGames());
            }
        }
    }
}
```

设计思路

- **棋盘显示：** 使用 buildLeftSection 方法构建棋盘的左侧部分，显示棋盘的行和列以及棋子状态。 使用 buildMiddleSection 方法显示当前游戏的分数和玩家信息。 使用 buildRightSection 方法显示游戏列表，方便玩家切换游戏。

- **游戏状态管理：**

判断当前游戏是否结束，并根据游戏类型（Reversi 或 Peace）显示胜负结果或平局信息。 如果所有游戏结束，提示玩家开始新游戏或退出。

- **玩家提示：** 根据当前玩家的状态，提示合法的输入选项（如移动、跳过回合、切换游戏等）。 如果当前玩家无法进行操作，提示其跳过回合。

- **清屏功能：** 使用 clearConsole 方法清除控制台，保持界面整洁。 根据操作系统选择适合的清屏命令（Windows 使用 cls，macOS/Linux 使用 ANSI 控制码）。

6. **GameManager 的游戏管理逻辑**

```
public static void start(Scanner scanner) {
    Player[] players = GameSetup.initializePlayers(scanner);

    // 初始化 Peace 和 Reversi 游戏
    games.add(new Peace(players[0], players[1], scanner));
    games.get(0).setGameID(0);
    games.add(new Reversi(players[0], players[1], scanner));
    games.get(1).setGameID(1);
    games.get(1).canPlacePiece(players[0].pieceType);
    currentGame = 0;

    // 显示初始棋盘
    GameView.printBoard(games.get(0), games.get(0).getBoard(), players[0],
players[0], players[1]);

    while (true) {
        GameEngine engine = games.get(currentGame);
        int[] input = InputUtils.readValidInput(scanner, engine,
players[engine.getCurrentPlayerIndice()].pieceType);

        // 处理退出游戏
        if (input[1] == -2) {
            System.out.println("Exiting the game.");
            break;
        }

        // 处理新游戏创建
        else if (input[1] == -4) {
            games.add(new Peace(players[0], players[1], scanner));
            games.get(games.size() - 1).setGameID(games.size() - 1);
            GameView.printBoard(engine, engine.getBoard(),
players[engine.getCurrentPlayerIndice()], players[0], players[1]);
        } else if (input[1] == -5) {
            games.add(new Reversi(players[0], players[1], scanner));
            games.get(games.size() - 1).setGameID(games.size() - 1);
```

```
            games.get(games.size() -
1).canPlacePiece(players[0].pieceType);
            GameView.printBoard(engine, engine.getBoard(),
players[engine.getCurrentPlayerIndice()], players[0], players[1]);
        }

        // 处理跳过回合
        else if (input[1] == -3) {
            if (engine instanceof Reversi) {
                if
(engine.canPlacePiece(players[engine.getCurrentPlayerIndice()].pieceType))
{
                    System.out.println("You can place a piece. Please try
again.");
                    continue;
                }

engine.setCurrentPlayerIndice((engine.getCurrentPlayerIndice() + 1) % 2);
                engine.PassCounterAdd();

engine.canPlacePiece(players[engine.getCurrentPlayerIndice()].pieceType);
            } else if (engine instanceof Peace) {
                System.out.println("You cannot pass in Peace mode. Please
try again.");
                continue;
            }
            GameView.printBoard(engine, engine.getBoard(),
players[engine.getCurrentPlayerIndice()], players[0], players[1]);
        }

        // 处理游戏切换
        else if (input[1] == -1) {
            if (input[0] >= 0 && input[0] < games.size()) {
                currentGame = input[0];
            } else {
                System.out.println("Invalid game number. Please try
again.");
                continue;
            }
            engine = games.get(currentGame);

engine.canPlacePiece(players[engine.getCurrentPlayerIndice()].pieceType);
            GameView.printBoard(engine, engine.getBoard(),
players[engine.getCurrentPlayerIndice()], players[0], players[1]);
        }

        // 处理棋子放置
        else {
            int col = input[0];
            int row = input[1];
            engine.placePiece(col, row,
players[engine.getCurrentPlayerIndice()].pieceType);
            engine.setCurrentPlayerIndice((engine.getCurrentPlayerIndice()
+ 1) % 2);
```

```
engine.canPlacePiece(players[engine.getCurrentPlayerIndice()].pieceType);
            GameView.printBoard(engine, engine.getBoard(),
players[engine.getCurrentPlayerIndice()], players[0], players[1]);
        }
    }
}
```

设计思路

- **游戏初始化：** 初始化玩家信息，并创建两个游戏（Peace 和 Reversi）。 设置每个游戏的 GameID，方便后续管理。
- **游戏循环：** 使用 while 循环处理玩家输入，根据输入执行对应的操作。
- **输入处理：** 调用 InputUtils.readValidInput 方法解析输入，支持多种操作：
- 退出游戏：处理 quit 命令，结束游戏。
- 创建新游戏：支持创建 Peace 或 Reversi 新游戏。
- 跳过回合：检查当前游戏是否允许跳过回合，并切换玩家。
- 切换游戏：根据输入的游戏编号切换到对应的游戏。
- 棋子放置：在合法位置放置棋子，并切换到下一位玩家。
- **游戏状态更新：** 每次操作后，调用 GameView.printBoard 更新棋盘显示，并提示玩家下一步操作。
- **错误处理：** 检查输入是否合法（如游戏编号是否有效、是否可以跳过回合等），并提示用户重新输入。

7. GameEngine 的游戏引擎接口

```
public interface GameEngine {
    Board board = new Board();
    public Player[] players = new Player[2];
    public Scanner scanner = new Scanner(System.in);
    public int currentPlayerIndice = 0;
    public int passCounter = 0;

    public int getPassCounter();

    public void PassCounterAdd();

    public static int GameID = 0;

    public int getCurrentPlayerIndice();

    public void setCurrentPlayerIndice(int currentPlayerIndice);

    public boolean isGameOver();
```

```
    public int getGameID();

    public void setGameID(int id);

    public boolean canPlacePiece(Piece piece);

    public void placePiece(int col, int row, Piece piece);

    public Board getBoard();

    public int getHowManyPieces(Piece piece);
}
```

设计思路

- **统一接口设计：** 定义游戏引擎的接口，为所有游戏（如 Reversi 和 Peace）提供统一的操作方法。 确保不同游戏可以通过相同的接口进行管理和操作。
- **核心方法：** getPassCounter 和 PassCounterAdd： 用于追踪玩家跳过回合的次数，特别是在 Reversi 中判断游戏是否结束。

getCurrentPlayerIndice 和 setCurrentPlayerIndice： 用于管理当前玩家的索引，实现玩家轮流操作。

isGameOver： 判断游戏是否结束，根据具体游戏的规则进行实现。

canPlacePiece 和 placePiece： 检查棋子是否可以放置，以及执行棋子放置的操作。

getHowManyPieces： 计算棋盘上某种类型棋子的数量，用于显示分数或判断胜负。

- **游戏管理支持：** getGameID 和 setGameID： 为每个游戏分配唯一的 ID，方便在多游戏模式下进行管理和切换。

getBoard： 提供对棋盘数据结构的访问，支持游戏逻辑和显示功能。

- **扩展性：** 通过接口的设计，未来可以轻松添加新的游戏类型，只需实现该接口即可。 确保代码的模块化和可维护性。

运行示例

定义玩家

```
Please enter the first player name (Using the black piece ●): Alice
Please enter the second player name (Using the white piece ○): Bob
```

初始状态

```
  A B C D E F G H
1 · · · · · · · ·
```

```
2 · · · · · · · ·                                       Game List
3 · · · · · · · ·         Current Game: 1               1.  Peace
4 · · · ● ○ · · ·         Player [Alice] ○              2.  Reversi
5 · · · ○ ● · · ·         Player [Bob]
6 · · · · · · · ·
7 · · · · · · · ·
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) :
```

peace运行示例

- 规则同lab2。双方轮流在空白处落子，直至棋盘已满。没有记分逻辑。

```
  A B C D E F G H
1 · · · · · · · ·
2 · · · · · · · ·                                       Game List
3 · · · · · · · ·         Current Game: 1               1.  Peace
4 · · · ● ○ · · ·         Player [Alice] ○              2.  Reversi
5 · · · ○ ● · · ·         Player [Bob]
6 · · · · · · · ·
7 · · · · · · · ·
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) : 1a
```

```
  A B C D E F G H
1 ○ · · · · · · ·
2 · · · · · · · ·                                       Game List
3 · · · · · · · ·         Current Game: 1               1.  Peace
4 · · · ● ○ · · ·         Player [Alice]                2.  Reversi
5 · · · ○ ● · · ·         Player [Bob]   ●
6 · · · · · · · ·
7 · · · · · · · ·
8 · · · · · · · ·
Player Bob, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) :
```

```
  A B C D E F G H
1 ○ · · · · · · ·
2 · · · · · · · ·                                       Game List
3 · · · · · · · ·         Current Game: 1               1.  Peace
4 · · · ● ○ · · ·         Player [Alice]                2.  Reversi
5 · · · ○ ● · · ·         Player [Bob]   ●
6 · · · · · · · ·
7 · · · · · · · ·
8 · · · · · · · ·
```

```
Player Bob, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) : 1b
```

```
  A B C D E F G H
1 ○ ● · · · · · ·
2 · · · · · · · ·                                       Game List
3 · · · · · · · ·         Current Game: 1               1.  Peace
4 · · · ● ○ · · ·         Player [Alice] ○              2.  Reversi
5 · · · ○ ● · · ·         Player [Bob]
6 · · · · · · · ·
7 · · · · · · · ·
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) :
```

- 能够应对非法输入

```
  A B C D E F G H
1 ○ ● · · · · · ·
2 · · · · · · · ·                                       Game List
3 · · · · · · · ·         Current Game: 1               1.  Peace
4 · · · ● ○ · · ·         Player [Alice] ○              2.  Reversi
5 · · · ○ ● · · ·         Player [Bob]
6 · · · · · · · ·
7 · · · · · · · ·
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) : 1a

This position cannot be placed. Please try again.
1k
Invalid input format. Please enter a number (1-8) followed by a letter (A-
H) / a board number (1-2) / new game (peace / reversi) / quit the game
(quit) : 9a
Invalid input format. Please enter a number (1-8) followed by a letter (A-
H) / a board number (1-2) / new game (peace / reversi) / quit the game
(quit) : abc
Invalid input format. Please enter a number (1-8) followed by a letter (A-
H) / a board number (1-2) / new game (peace / reversi) / quit the game
(quit) :
```

切换游戏（1->2）

```
  A B C D E F G H
1 ○ ● · · · · · ·
2 · · · · · · · ·                                       Game List
3 · · · · · · · ·         Current Game: 1               1.  Peace
```

```
4 · · · ● ○ · · ·          Player [Alice] ○              2.  Reversi
5 · · · ○ ● · · ·          Player [Bob]
6 · · · · · · · ·
7 · · · · · · · ·
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) : 1a

This position cannot be placed. Please try again.
1k
Invalid input format. Please enter a number (1-8) followed by a letter (A-
H) / a board number (1-2) / new game (peace / reversi) / quit the game
(quit) : 9a
Invalid input format. Please enter a number (1-8) followed by a letter (A-
H) / a board number (1-2) / new game (peace / reversi) / quit the game
(quit) : abc
Invalid input format. Please enter a number (1-8) followed by a letter (A-
H) / a board number (1-2) / new game (peace / reversi) / quit the game
(quit) : 2
```

```
  A B C D E F G H
1 · · · · · · · ·
2 · · · · · · · ·                                      Game List
3 · · · + · · · ·          Current Game: 2             1.  Peace
4 · · + ● ○ · · ·          Player [Alice] ○  2         2.  Reversi
5 · · · ○ ● + · ·          Player [Bob]      2
6 · · · · + · · ·
7 · · · · · · · ·
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) :
```

reversi运行示例

```
  A B C D E F G H
1 · · · · · · · ·
2 · · · · · · · ·                                      Game List
3 · · · + · · · ·          Current Game: 2             1.  Peace
4 · · + ● ○ · · ·          Player [Alice] ○  2         2.  Reversi
5 · · · ○ ● + · ·          Player [Bob]      2
6 · · · · + · · ·
7 · · · · · · · ·
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) : 3d
```

```
  A B C D E F G H
1 · · · · · · · ·
2 · · · · · · · ·                                   Game List
3 · · + ○ + · · ·         Current Game: 2           1.  Peace
4 · · · ○ ○ · · ·         Player [Alice]    4       2.  Reversi
5 · · + ○ ● · · ·         Player [Bob]   ●  1
6 · · · · · · · ·
7 · · · · · · · ·
8 · · · · · · · ·
Player Bob, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) :
```

```
  A B C D E F G H
1 · · · · · · · ·
2 · · · · · · · ·                                   Game List
3 · · + ○ + · · ·         Current Game: 2           1.  Peace
4 · · · ○ ○ · · ·         Player [Alice]    4       2.  Reversi
5 · · + ○ ● · · ·         Player [Bob]   ●  1
6 · · · · · · · ·
7 · · · · · · · ·
8 · · · · · · · ·
Player Bob, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) : 3c
```

```
  A B C D E F G H
1 · · · · · · · ·
2 · · · · · · · ·                                   Game List
3 · + ● ○ · · · ·         Current Game: 2           1.  Peace
4 · · + ● ○ · · ·         Player [Alice] ○  3       2.  Reversi
5 · · · ○ ● + · ·         Player [Bob]      3
6 · · · · + · · ·
7 · · · · · · · ·
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) :
```

- 能够应对非法输入

```
  A B C D E F G H
1 · · · · · · · ·
2 · · · · · · · ·                                   Game List
3 · + ● ○ · · · ·         Current Game: 2           1.  Peace
4 · · + ● ○ · · ·         Player [Alice] ○  3       2.  Reversi
5 · · · ○ ● + · ·         Player [Bob]      3
6 · · · · + · · ·
7 · · · · · · · ·
```

```
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) : 1a

This position cannot be placed. Please try again.
3c

This position cannot be placed. Please try again.
1j
Invalid input format. Please enter a number (1-8) followed by a letter (A-
H) / a board number (1-2) / new game (peace / reversi) / quit the game
(quit) : 9a
Invalid input format. Please enter a number (1-8) followed by a letter (A-
H) / a board number (1-2) / new game (peace / reversi) / quit the game
(quit) : pass

You can place a piece. Please try again.
abc
Invalid input format. Please enter a number (1-8) followed by a letter (A-
H) / a board number (1-2) / new game (peace / reversi) / quit the game
(quit) :
```

添加新游戏

```
  A B C D E F G H
1 · · · · · · · ·
2 · · · · · · · ·                                   Game List
3 · + ● ○ · · · ·         Current Game: 2           1.  Peace
4 · · + ● ○ · · ·         Player [Alice] ○  3       2.  Reversi
5 · · · ○ ● + · ·         Player [Bob]      3
6 · · · · + · · ·
7 · · · · · · · ·
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) : peace
```

```
  A B C D E F G H
1 · · · · · · · ·
2 · · · · · · · ·                                   Game List
3 · + ● ○ · · · ·         Current Game: 2           1.  Peace
4 · · + ● ○ · · ·         Player [Alice] ○  3       2.  Reversi
5 · · · ○ ● + · ·         Player [Bob]      3       3.  Peace
6 · · · · + · · ·
7 · · · · · · · ·
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-3) / new
game (peace / reversi) / quit the game (quit) :
```

```
  A B C D E F G H
1 · · · · · · · ·
2 · · · · · · · ·                                   Game List
3 · + ● ○ · · · ·         Current Game: 2           1.  Peace
4 · · + ● ○ · · ·         Player [Alice] ○  3       2.  Reversi
5 · · · ○ ● + · ·         Player [Bob]      3       3.  Peace
6 · · · · + · · ·
7 · · · · · · · ·
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-3) / new
game (peace / reversi) / quit the game (quit) : reversi
```

```
  A B C D E F G H
1 · · · · · · · ·
2 · · · · · · · ·                                   Game List
3 · + ● ○ · · · ·         Current Game: 2           1.  Peace
4 · · + ● ○ · · ·         Player [Alice] ○  3       2.  Reversi
5 · · · ○ ● + · ·         Player [Bob]      3       3.  Peace
6 · · · · + · · ·                                   4.  Reversi
7 · · · · · · · ·
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-4) / new
game (peace / reversi) / quit the game (quit) :
```

- 切换到新游戏

```
  A B C D E F G H
1 · · · · · · · ·
2 · · · · · · · ·                                   Game List
3 · + ● ○ · · · ·         Current Game: 2           1.  Peace
4 · · + ● ○ · · ·         Player [Alice] ○  3       2.  Reversi
5 · · · ○ ● + · ·         Player [Bob]      3       3.  Peace
6 · · · · + · · ·                                   4.  Reversi
7 · · · · · · · ·
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-4) / new
game (peace / reversi) / quit the game (quit) : 3
```

```
  A B C D E F G H
1 · · · · · · · ·
2 · · · · · · · ·                                   Game List
3 · · · · · · · ·         Current Game: 3           1.  Peace
4 · · · ● ○ · · ·         Player [Alice] ○          2.  Reversi
5 · · · ○ ● · · ·         Player [Bob]              3.  Peace
6 · · · · · · · ·                                   4.  Reversi
7 · · · · · · · ·
```

```
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-4) / new
game (peace / reversi) / quit the game (quit) :
```

```
  A B C D E F G H
1 · · · · · · · ·
2 · · · · · · · ·                                   Game List
3 · · · · · · · ·         Current Game: 3           1.  Peace
4 · · · ● ○ · · ·         Player [Alice] ○          2.  Reversi
5 · · · ○ ● · · ·         Player [Bob]              3.  Peace
6 · · · · · · · ·                                   4.  Reversi
7 · · · · · · · ·
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-4) / new
game (peace / reversi) / quit the game (quit) : 4
```

```
  A B C D E F G H
1 · · · · · · · ·
2 · · · · · · · ·                                   Game List
3 · · · + · · · ·         Current Game: 4           1.  Peace
4 · · + ● ○ · · ·         Player [Alice] ○  2       2.  Reversi
5 · · · ○ ● + · ·         Player [Bob]      2       3.  Peace
6 · · · · + · · ·                                   4.  Reversi
7 · · · · · · · ·
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-4) / new
game (peace / reversi) / quit the game (quit) :
```

- 再切换回现有游戏

```
  A B C D E F G H
1 · · · · · · · ·
2 · · · · · · · ·                                   Game List
3 · · · + · · · ·         Current Game: 4           1.  Peace
4 · · + ● ○ · · ·         Player [Alice] ○  2       2.  Reversi
5 · · · ○ ● + · ·         Player [Bob]      2       3.  Peace
6 · · · · + · · ·                                   4.  Reversi
7 · · · · · · · ·
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-4) / new
game (peace / reversi) / quit the game (quit) : 1
```

```
  A B C D E F G H
1 ○ ● · · · · · ·
2 · · · · · · · ·                                   Game List
```

```
3 · · · · · · · · ·          Current Game: 1                1.  Peace
4 · · · ● ○ · · · ·          Player [Alice] ○               2.  Reversi
5 · · · ○ ● · · · ·          Player [Bob]                   3.  Peace
6 · · · · · · · · ·                                         4.  Reversi
7 · · · · · · · · ·
8 · · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-4) / new
game (peace / reversi) / quit the game (quit) :
```

```
  A B C D E F G H
1 ○ ● · · · · · ·
2 · · · · · · · ·                                           Game List
3 · · · · · · · ·          Current Game: 1                1.  Peace
4 · · · ● ○ · · ·          Player [Alice] ○               2.  Reversi
5 · · · ○ ● · · ·          Player [Bob]                   3.  Peace
6 · · · · · · · ·                                         4.  Reversi
7 · · · · · · · ·
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-4) / new
game (peace / reversi) / quit the game (quit) : 2
```

```
  A B C D E F G H
1 · · · · · · · ·
2 · · · · · · · ·                                           Game List
3 · + ● ○ · · · ·          Current Game: 2                1.  Peace
4 · · + ● ○ · · ·          Player [Alice] ○  3            2.  Reversi
5 · · · ○ ● + · ·          Player [Bob]      3            3.  Peace
6 · · · · + · · ·                                         4.  Reversi
7 · · · · · · · ·
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-4) / new
game (peace / reversi) / quit the game (quit) :
```

单局游戏的结束逻辑

- peace结束的条件：棋盘下满。

```
  A B C D E F G H
1 ○ ● ○ ○ ○ ○ ○ ○
2 ● ○ ● ● ● ● ● ●                                           Game List
3 ○ ● ○ ○ ○ ○ ○ ○          Current Game: 1                1.  Peace
4 ● ○ ● ● ○ ● ● ●          Player [Alice]                 2.  Reversi
5 ○ ● ○ ○ ● ○ ○ ○          Player [Bob]   ●               3.  Peace
6 ● ○ ● ● ● ● ● ●                                         4.  Reversi
7 ○ ● ○ ○ ○ ○ ○ ○
8 ● ○ ● ● ● ● ● ·
```

```
Player Bob, please enter your move (1-8,a-h) / game number (1-4) / new
game (peace / reversi) / quit the game (quit) : 8h
```

```
  A B C D E F G H
1 ○ ● ○ ○ ○ ○ ○ ○
2 ● ○ ● ● ● ● ● ●                                   Game List
3 ○ ● ○ ○ ○ ○ ○ ○          Current Game: 1          1.  Peace
4 ● ○ ● ● ○ ● ● ●          Player [Alice] ○         2.  Reversi
5 ○ ● ○ ○ ● ○ ○ ○          Player [Bob]             3.  Peace
6 ● ○ ● ● ● ● ● ●                                   4.  Reversi
7 ○ ● ○ ○ ○ ○ ○ ○
8 ● ○ ● ● ● ● ● ●
Game 1 is over now.
Please enter another board number to continue / new game (peace / reversi)
/ quit the game (quit) :
```

- reversi结束的条件：棋盘下满，或者双方均无合法落子位置。 为方便测验，改为4*4的棋盘

```
  A B C D E F G H
1 · + · ·
2 + ● ○ ·                                           Game List
3 · ○ ● +                  Current Game: 2          1.  Peace
4 · · + ·                  Player [Alice] ○  2      2.  Reversi
Player Alice, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) :
```

```
  A B C D E F G H
1 · + · ·
2 + ● ○ ·                                           Game List
3 · ○ ● +                  Current Game: 2          1.  Peace
4 · · + ·                  Player [Alice] ○  2      2.  Reversi
Player Alice, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) : 1b
```

```
  A B C D E F G H
1 + ○ + ·
2 · ○ ○ ·                                           Game List
3 + ○ ● ·                  Current Game: 2          1.  Peace
4 · · · ·                  Player [Alice]    4      2.  Reversi
Player Bob, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) :
```

```
  A B C D E F G H
1 ● ○ · ·
2 ● ○ ○ ○                                          Game List
3 ● ● ○ ·                Current Game: 2           1.  Peace
4 ● ● ● ·                Player [Alice] ○  5       2.  Reversi
Player Alice, you do not have place for your piece, pleace enter pass /
game number (1-2) / new game (peace / reversi) / quit :
```

```
  A B C D E F G H
1 ● ○ · ·
2 ● ○ ○ ○                                          Game List
3 ● ● ○ ·                Current Game: 2           1.  Peace
4 ● ● ● ·                Player [Alice] ○  5       2.  Reversi
Player Alice, you do not have place for your piece, pleace enter pass /
game number (1-2) / new game (peace / reversi) / quit : pass
```

```
  A B C D E F G H
1 ● ○ + +
2 ● ○ ○ ○                                          Game List
3 ● ● ○ +                Current Game: 2           1.  Peace
4 ● ● ● +                Player [Alice]    5       2.  Reversi
Player Bob, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) :
```

```
  A B C D E F G H
1 ● ○ + +
2 ● ○ ○ ○                                          Game List
3 ● ● ○ +                Current Game: 2           1.  Peace
4 ● ● ● +                Player [Alice]    5       2.  Reversi
Player Bob, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) : 4d
```

```
  A B C D E F G H
1 ● ○ · ·
2 ● ● ○ ○                                          Game List
3 ● ● ● ·                Current Game: 2           1.  Peace
4 ● ● ● ●                Player [Alice] ○  3       2.  Reversi
Player Alice, you do not have place for your piece, pleace enter pass /
game number (1-2) / new game (peace / reversi) / quit :
```

```
  A B C D E F G H
1 ● ○ + ●
```

```
2 ● ● ○ ○                                              Game List
3 ● ● ● ○                  Current Game: 2             1.  Peace
4 ● ● ● ●                  Player [Alice]    4         2.  Reversi
Player Bob, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) : 1c
```

```
  A B C D E F G H
1 ● ● ● ●
2 ● ● ● ○                                              Game List
3 ● ● ● ○                  Current Game: 2             1.  Peace
4 ● ● ● ●                  Player [Alice] ○  2         2.  Reversi
Game 2 is over now.
Player [Alice] ○    2
Player [Bob] ●    14
Player [Bob]  wins!
Please enter another board number to continue / new game (peace / reversi)
/ quit the game (quit) :
```

- 双方均无合法落子位置。

```
  A B C D E F G H
1 ● ● ● ●
2 ● ● ● ●                                              Game List
3 ● ● ● ●                  Current Game: 2             1.  Peace
4 · ○ · ·                  Player [Alice] ○  1         2.  Reversi
Player Alice, you do not have place for your piece, pleace enter pass /
game number (1-2) / new game (peace / reversi) / quit :
```

```
  A B C D E F G H
1 ● ● ● ●
2 ● ● ● ●                                              Game List
3 ● ● ● ●                  Current Game: 2             1.  Peace
4 · ○ · ·                  Player [Alice] ○  1         2.  Reversi
Player Alice, you do not have place for your piece, pleace enter pass /
game number (1-2) / new game (peace / reversi) / quit : pass
```

```
  A B C D E F G H
1 ● ● ● ●
2 ● ● ● ●                                              Game List
3 ● ● ● ●                  Current Game: 2             1.  Peace
4 · ○ · ·                  Player [Alice]    1         2.  Reversi
Player Bob, you do not have place for your piece, pleace enter pass / game
number (1-2) / new game (peace / reversi) / quit :
```

```
  A B C D E F G H
1 ● ● ● ●
2 ● ● ● ●                                          Game List
3 ● ● ● ●                Current Game: 2           1.  Peace
4 · ○ · ·                Player [Alice]    1       2.  Reversi
Player Bob, you do not have place for your piece, pleace enter pass / game
number (1-2) / new game (peace / reversi) / quit : pass
```

```
  A B C D E F G H
1 ● ● ● ●
2 ● ● ● ●                                          Game List
3 ● ● ● ●                Current Game: 2           1.  Peace
4 · ○ · ·                Player [Alice] ○  1       2.  Reversi
Game 2 is over now.
Player [Alice] ○    1
Player [Bob] ●    12
Player [Bob]  wins!
Please enter another board number to continue / new game (peace / reversi)
/ quit the game (quit) :
```

即便游戏结束，依然可以切换过去查看游戏结果。

```
  A B C D E F G H
1 ○ ○ ○ ○ ○ ○ ○ ○
2 ● ● ● ● ● ● ● ●                                  Game List
3 ○ ○ ○ ○ ○ ○ ○ ○        Current Game: 1           1.  Peace
4 ● ● ● ● ○ ● ● ●        Player [Alice] ○          2.  Reversi
5 ○ ○ ○ ○ ● ○ ○ ○        Player [Bob]
6 ● ● ● ● ● ● ● ●
7 ○ ○ ○ ○ ○ ○ ○ ○
8 ● ● ● ● ● ● ● ●
Game 1 is over now.
Please enter another board number to continue / new game (peace / reversi)
/ quit the game (quit) : 2
```

```
  A B C D E F G H
1 · · · · · · · ·
2 · · · · · · · ·                                  Game List
3 · + ● ○ · · · ·        Current Game: 2           1.  Peace
4 · · + ● ○ · · ·        Player [Alice] ○  3       2.  Reversi
5 · · · ○ ● + · ·        Player [Bob]      3
6 · · · · + · · ·
7 · · · · · · · ·
8 · · · · · · · ·
```

```
Player Alice, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) :
```

```
  A B C D E F G H
1 · · · · · · · ·
2 · · · · · · · ·                                   Game List
3 · + ● ○ · · · ·         Current Game: 2           1.  Peace
4 · · + ● ○ · · ·         Player [Alice] ○  3       2.  Reversi
5 · · · ○ ● + · ·         Player [Bob]      3
6 · · · · + · · ·
7 · · · · · · · ·
8 · · · · · · · ·
Player Alice, please enter your move (1-8,a-h) / game number (1-2) / new
game (peace / reversi) / quit the game (quit) : 1
```

```
 A B C D E F G H
1 ○ ○ ○ ○ ○ ○ ○ ○
2 ● ● ● ● ● ● ● ●                                   Game List
3 ○ ○ ○ ○ ○ ○ ○ ○         Current Game: 1           1.  Peace
4 ● ● ● ● ○ ● ● ●         Player [Alice] ○          2.  Reversi
5 ○ ○ ○ ○ ● ○ ○ ○         Player [Bob]
6 ● ● ● ● ● ● ● ●
7 ○ ○ ○ ○ ○ ○ ○ ○
8 ● ● ● ● ● ● ● ●
Game 1 is over now.
Please enter another board number to continue / new game (peace / reversi)
/ quit the game (quit) :
```

玩家输入quit以退出程序。

```
 A B C D E F G H
1 ○ ○ ○ ○ ○ ○ ○ ○
2 ● ● ● ● ● ● ● ●                                   Game List
3 ○ ○ ○ ○ ○ ○ ○ ○         Current Game: 1           1.  Peace
4 ● ● ● ● ○ ● ● ●         Player [Alice] ○          2.  Reversi
5 ○ ○ ○ ○ ● ○ ○ ○         Player [Bob]
6 ● ● ● ● ● ● ● ●
7 ○ ○ ○ ○ ○ ○ ○ ○
8 ● ● ● ● ● ● ● ●
Game 1 is over now.
Please enter another board number to continue / new game (peace / reversi)
/ quit the game (quit) : quit
```

```
  A B C D E F G H
1 ○ ○ ○ ○ ○ ○ ○ ○
2 ● ● ● ● ● ● ● ●                                   Game List
3 ○ ○ ○ ○ ○ ○ ○ ○         Current Game: 1           1.  Peace
4 ● ● ● ● ○ ● ● ●         Player [Alice] ○          2.  Reversi
5 ○ ○ ○ ○ ● ○ ○ ○         Player [Bob]
6 ● ● ● ● ● ● ● ●
7 ○ ○ ○ ○ ○ ○ ○ ○
8 ● ● ● ● ● ● ● ●
Game 1 is over now.
Please enter another board number to continue / new game (peace / reversi)
/ quit the game (quit) : quit
Exiting the game.
(base) charleslam@LamdeMacBook-Pro lab2 %
```