## Playing Othello

The game begins with four discs placed in a square in the middle of the grid, two with their black side facing up, two with their white side facing up, with the two discs with the same colour on a diagonal with each other. The convention is that the discs with the black side facing up are to the north-east and south-west as both players look at the board. Figure 1 shows the starting position of a game of Othello[1].

```
    --- --- --- --- --- --- --- ---
 1 |   |   |   |   |   |   |   |   |
    --- --- --- --- --- --- --- ---
 2 |   |   |   |   |   |   |   |   |
    --- --- --- --- --- --- --- ---
 3 |   |   |   |   |   |   |   |   |
    --- --- --- --- --- --- --- ---
 4 |   |   |   | W | B |   |   |   |
    --- --- --- --- --- --- --- ---
 5 |   |   |   | B | W |   |   |   |
    --- --- --- --- --- --- --- ---
 6 |   |   |   |   |   |   |   |   |
    --- --- --- --- --- --- --- ---
 7 |   |   |   |   |   |   |   |   |
    --- --- --- --- --- --- --- ---
 8 |   |   |   |   |   |   |   |   |
    --- --- --- --- --- --- --- ---
     a   b   c   d   e   f   g   h
```

Players take alternate turns. If one player cannot make a valid move (ie they cannot place a disc to capture at least one disc of the other colour), they must “pass” and play passes back to the other player. When neither player can move, the game ends. This occurs when the grid has filled up (all 64 discs have been placed) or when neither player can legally place a disc in any of the remaining squares. In the latter case, it means the game may end before the board is completely filled. This may occur because one player has no discs remaining on the board in their colour or because neither player can place a disc to capture at least one disc of the other colour. A player cannot pass if they have a valid move, even if that move disadvantages them.

The winner of the game is the player who has more discs on the board when the game ends. A draw or tie is possible.

### Program functionality

Your program should have the following functionality:

- Ask each player to provide their name as input from the keyboard and assign this name to the player;
- Assign players a disc colour;
- Keep track of each player's score;
- Initialise the components of the game, eg initialise the board, initialise the score, set Black to be the first player to move;
- Request a move from each player in turn. A move should be entered at the keyboard. A move can either be **p** (for "Pass") or in the format `ld`, where `l` is a letter in the range **a** to **h**, and `d` is a digit in the range **1** to **8**;

  rows

  columns
- Check that the entered move is valid and, if it is, effect the move. If the move is not valid, request another move;
- A player should only be allowed to Pass if they have no valid move;
- Update and print the board after each move;
- When the game is over, establish which player has won, print out the winner, the score and the final board. The date and time of the game and the result of the game (ie the winner, the score) should also be appended to a file `othello-results.txt`.

Note that the purpose of your program is to "manage" a game between two human players, *not* to develop playing strategies whereby the program can play against a human player.

Also, it is *not* necessary to develop a graphical interface: all that is required is to take as input a player's move from the keyboard and to print the board out on the screen using characters and strings. An example of the output to the screen is presented on the last page of this specification.

## Program code

You should use structures to represent elements of the game, eg the board, the players, etc. For example, the struct for the board would include a 2-Dimensional array, as well as the score and information on the next player to play.

## Code requirements

- Use functions where possible;
- Use data structures to represent the board, the players, etc;
- Break your code into independent modules;
- Comment your code;
- Use a Git repository;
- In your repository include a text file describing:
  - How you decided to implement the board, the players, the disks. . .
  - How you decided to implement the game logic
  - . . .

## Evaluation Criteria

Your solution will be graded according to the following criteria:

- Your code should be well commented and appropriately divided into modules;
- Your work should be placed in the Git repository;
- The submitted text file should describe your design choices;
- Implementation of the players;
- Implementation of the board;
- Implementation of the game logic;
- Initialisation of the game elements;
- Correct printing of the board;
- Correct determination of the winner;
- Writing of result of game to file on disk.

## Example

The following is an example of how the game should start:

```
$ othello
        ***  Welcome to Othello!  ***

Enter name of Player 1 (Black): John

Enter name of Player 2 (White): Tigger

        Score: John (Black) 2:2 Tigger (White)
             --- --- --- --- --- --- --- ---
        1   |   |   |   |   |   |   |   |   |
             --- --- --- --- --- --- --- ---
        2   |   |   |   |   |   |   |   |   |
             --- --- --- --- --- --- --- ---
        3   |   |   |   |   |   |   |   |   |
             --- --- --- --- --- --- --- ---
        4   |   |   |   | W | B |   |   |   |
             --- --- --- --- --- --- --- ---
        5   |   |   |   | B | W |   |   |   |
             --- --- --- --- --- --- --- ---
        6   |   |   |   |   |   |   |   |   |
             --- --- --- --- --- --- --- ---
        7   |   |   |   |   |   |   |   |   |
             --- --- --- --- --- --- --- ---
        8   |   |   |   |   |   |   |   |   |
             --- --- --- --- --- --- --- ---
              a   b   c   d   e   f   g   h
```