The Problem

We are going to be implementing a version of the classic children's game Connect 4 except that in our version we will allow the user to select the size of the board as well as the number of pieces to win.

Step 1: Learn About Connect 4

In Connect 4 two players take turns dropping pieces into a 6 X 7 board attempting to get 4 or more pieces in a row either vertically, horizontally, or diagonally to win.

If you have never played Connect 4 play a few games to understand how the game works. You can play a version of Connect 4 here. Note that you will not have to make this exact game and there will be differences between this game and the version we make. This is just to help you understand how things work.

Our Version of the Game: ConnectN

ConnectN is played exactly like Connect 4 except that the users can specify the dimensions of the board as well as how many places to win. Also, unlike the version I linked to above, we will always have 2 humans play against each other instead of having humans play against the computer.

Requirements

Classes

Your program must have at least the following classes and use them in a meaningful way

- Game
- Player
- Board

The names of the classes must be exactly as stated above.

Each class should appear in its own appropriately named file.

You can, of course, have **more** classes (I had one additional class) but you must have at least the above.

Testing

You must provide unit tests to sufficiently exercise your code and verify that it is functioning correctly.

Not everything will need/should have a test. For example, you should not have a test to see if the entire program runs correctly from start to finish because that is an end to end test and not a unit test. And while you could write one it would be extremely complicated and hard to do in unittest as it is not designed to do that. Instead, write tests for the individual methods and see if they work correctly.

Some of your tests may require that you check what is printed to the screen. I have created a class called PrintCapturer that when used with `patch` can be used to capture the output of `print`. You can find this class in the test folder along with some examples of using it.

Type Hinting

Your code must be type hinted and have 5 or fewer errors when run by mypy with the following options enabled

- `--disallow-untyped-defs`
- `--disallow-incomplete-defs`
- `--check-untyped-defs`

Formatting

In order to help develop a good programming style, we will be running pylint on your submission. To pass the formatting test on Mimir your submission must be rated at an 8 or higher by pylint.

Command Line Arguments

Your program should accept 1 command line argument, the path to the configuration file.

The path to the file will always be given and will always be a valid path.

Configuration File

The configuration file contains the dimensions of the board as well as the number of pieces to win.

The contents of the configuration file will always be valid.

Configuration File Format

The configuration file has the following format

```
num_rows : integer
num_cols : integer
num_pieces_to_win : integer
blank_char : character
num_rows : integer
num_cols : integer
num_pieces_to_win : integer
blank_char : character
```

The order of the parameters can appear within any order of the file but they will always have the same keys.

Here is an example of what the configuration file would look like if we were trying to make the game of Connect 4.

```
num_rows : 6
num_cols : 7
num_pieces_to_win : 4
blank_char : *
num_rows : 6
num_cols : 7
num_pieces_to_win : 4
blank_char : *
```

And here is one that would create a 10 X 10 board with 5 pieces needed to win

```
blank_char : !
num_rows : 10
num_pieces_to_win : 5
num_cols : 10
blank_char : !
num_rows : 10
num_pieces_to_win : 5
```

`num_cols : 10`

Standard Input

Most of the input of the program will be given through standard input and consists of two main parts

1. Getting the information about the players
2. Playing the game

The input the players give you will **NOT** always be valid. If invalid input is given you should inform the user why what they entered is wrong and then prompt them again for correct input. This continues until the user finally enters valid input.

Player Input

When a player is created you should ask them for their name as well as the piece that they would like to use to represent themselves on the board. If either input ends up being invalid you should ask for valid input starting with the player's name again.

Player Name

- Prompt: `'Player {X} enter your name: '` where X is the player's number starting from 1

A player's name is valid if

- It contains at least 1 non-whitespace character
    - Error message: `'Your name cannot be the empty string or whitespace.'`
- Is not being used by the other player in the game
    - Checking for the same name should be case insensitive so if player 1 was named BOB player 2 could not use bob for their name
    - Error message: `'You cannot use {name} for your name as someone else is already using it.'`

Player Piece

- Prompt: 'Player {X} enter your piece: ' here X is the players number starting from 1

A player's piece is valid if

- It is not the empty string or whitespace

    - Error message: `'Your piece cannot be the empty string or whitespace.'`
- It is exactly 1 character long
    - Error message: `'{piece} is not a single character. Your piece can only be a single character.'`
- It is not the same as the board's blank character
    - Error message: `'Your piece cannot be the same as the blank character.'`
- It is not the same as the other player's piece
    - Error message: `'You cannot use {piece} for your piece as {player} is already using it.'`

Playing the Game

On each player's turn, you should ask them which column they would like to play their piece in.

- Prompt: `'{name}, please enter the column you want to play in: '`

Input is valid if

- The user enters a number
    - Error message: `'{maker}, column needs to be an integer. {str_move} is not an integer. '`
- That number corresponds to a column inside of the board
    - Error message: `'Your column needs to be between 0 and {num_cols - 1} but is actually {column}.`
- That column is not full
    - Error message: `'You cannot play in {column} because it is full.'`

What to Submit

A zip file that contains a folder named ConnectN with the following structure

- ConnectN
    - ConnectNGame
        - __init__.py
        - src
            - __init__.py
            - board.py
            - game.py
            - player.py
            - Any other source files you have
        - test

        - __init__.py
        - test_board.py
        - test_game.py
        - test_player.py
        - Any other test files you have
    - main.py

How Your Program Will Be Run

```
python3 ConnectN/main.py path_to_config_file
```

Hints and Tips

- Start early.
    - This assignment will take you longer than you think.
    - You will only have enough time to finish it if you start right away.
    - You do not know what problems you will run into until you run into them, so start early to give yourself enough time to overcome them and get help on them.
- Create an outline of your program before you even start writing code. Identify what your classes are going to be, what you want those classes to be able to do, and how they are likely going to interact.
    - Come to office hours **early** and discuss your design/organization. This help will go a long way to smoothing out the difficulty of this assignment.
- Try programming top down. You'll never know if you like it until you try it and you might find that it is easier than bottom-up.

Example Run

```
Player 1 enter your name: qq
Player 1 enter your piece: g
Player 2 enter your name:
Your name cannot be the empty string or whitespace.
Player 2 enter your name: qq
You cannot use qq for your name as someone else is already using it.
Player 2 enter your name: Bib
Player 2 enter your piece:
Your piece cannot be the empty string or whitespace.
Player 2 enter your name: Bib
Player 2 enter your piece: dog
dog is not a single character. Your piece can only be a single character.
```

```
Player 2 enter your name: Bib
Player 2 enter your piece: g
You cannot use g for your piece as qq is already using it.
Player 2 enter your name: Bib
Player 2 enter your piece: B
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
2 * * * * * * *
3 * * * * * * *
4 * * * * * * *
5 * * * * * * *
qq, please enter the column you want to play in: 4
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
2 * * * * * * *
3 * * * * * * *
4 * * * * * * *
5 * * * * g * *
Bib, please enter the column you want to play in: 3
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
2 * * * * * * *
3 * * * * * * *
4 * * * * * * *
5 * * * B g * *
qq, please enter the column you want to play in: d
qq, column needs to be an integer. d is not an integer.
qq, please enter the column you want to play in: .
qq, column needs to be an integer. . is not an integer.
qq, please enter the column you want to play in:
qq, column needs to be an integer.      is not an integer.
qq, please enter the column you want to play in: 1
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
2 * * * * * * *
3 * * * * * * *
4 * * * * * * *
5 * g * B g * *
Bib, please enter the column you want to play in: 3
  0 1 2 3 4 5 6
0 * * * * * * *
```

```
1 * * * * * * *
2 * * * * * * *
3 * * * * * * *
4 * * * B * * *
5 * g * B g * *
qq, please enter the column you want to play in: 4
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
2 * * * * * * *
3 * * * * * * *
4 * * * B g * *
5 * g * B g * *
Bib, please enter the column you want to play in: 4
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
2 * * * * * * *
3 * * * * B * *
4 * * * B g * *
5 * g * B g * *
qq, please enter the column you want to play in: 6
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
2 * * * * * * *
3 * * * * B * *
4 * * * B g * *
5 * g * B g * g
Bib, please enter the column you want to play in: 3
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
2 * * * * * * *
3 * * * B B * *
4 * * * B g * *
5 * g * B g * g
qq, please enter the column you want to play in: 1
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
2 * * * * * * *
3 * * * B B * *
4 * g * B g * *
5 * g * B g * g
```

```
Bib, please enter the column you want to play in: 3
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
2 * * * B * * *
3 * * * B B * *
4 * g * B g * *
5 * g * B g * g
Bib won the game!
Player 1 enter your name: qq
Player 1 enter your piece: g
Player 2 enter your name:
Your name cannot be the empty string or whitespace.
Player 2 enter your name: qq
You cannot use qq for your name as someone else is already using it.
Player 2 enter your name: Bib
Player 2 enter your piece:
Your piece cannot be the empty string or whitespace.
Player 2 enter your name: Bib
Player 2 enter your piece: dog
dog is not a single character. Your piece can only be a single character.
Player 2 enter your name: Bib
Player 2 enter your piece: g
You cannot use g for your piece as qq is already using it.
Player 2 enter your name: Bib
Player 2 enter your piece: B
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
2 * * * * * * *
3 * * * * * * *
4 * * * * * * *
5 * * * * * * *
qq, please enter the column you want to play in: 4
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
2 * * * * * * *
3 * * * * * * *
4 * * * * * * *
5 * * * * g * *
Bib, please enter the column you want to play in: 3
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
```

```
2 * * * * * * *
3 * * * * * * *
4 * * * * * * *
5 * * * B g * *
qq, please enter the column you want to play in: d
qq, column needs to be an integer. d is not an integer.
qq, please enter the column you want to play in: .
qq, column needs to be an integer. . is not an integer.
qq, please enter the column you want to play in:
qq, column needs to be an integer.      is not an integer.
qq, please enter the column you want to play in: 1
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
2 * * * * * * *
3 * * * * * * *
4 * * * * * * *
5 * g * B g * *
Bib, please enter the column you want to play in: 3
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
2 * * * * * * *
3 * * * * * * *
4 * * * B * * *
5 * g * B g * *
qq, please enter the column you want to play in: 4
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
2 * * * * * * *
3 * * * * * * *
4 * * * B g * *
5 * g * B g * *
Bib, please enter the column you want to play in: 4
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
2 * * * * * * *
3 * * * * B * *
4 * * * B g * *
5 * g * B g * *
qq, please enter the column you want to play in: 6
  0 1 2 3 4 5 6
0 * * * * * * *
```

```
1 * * * * * * *
2 * * * * * * *
3 * * * * B * *
4 * * * B g * *
5 * g * B g * g
Bib, please enter the column you want to play in: 3
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
2 * * * * * * *
3 * * * B B * *
4 * * * B g * *
5 * g * B g * g
qq, please enter the column you want to play in: 1
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
2 * * * * * * *
3 * * * B B * *
4 * g * B g * *
5 * g * B g * g
Bib, please enter the column you want to play in: 3
  0 1 2 3 4 5 6
0 * * * * * * *
1 * * * * * * *
2 * * * B * * *
3 * * * B B * *
4 * g * B g * *
5 * g * B g * g
Bib won the game!
```