ATCS 2021-2022
Ms. Namasivayam

Tic Tac Toe

In this project, you will build a working version of Tic Tac Toe (for the console) where a user can play against an AI. Each part of the project will bring you one step closer to completing the game and builds off of the previous part.

Part I - User vs User

A) **Build the Board**

You will construct the board that will be used to play Tic Tac Toe. A Tic Tac Toe board is a 3 x 3 board where each space is filled with a `-`.

Use the variable board that's already been provided to create a 3 x 3 2D list where all values consist of `-`.

Then, create a function `print_board()` that prints the board to the console. The printed board should also include the numerical value of each row and column on the board.

Finally, create a function `print_instructions()` that prints the instructions to the Tic Tac Toe game.

Your finished product should look something like this:

```
Welcome to TicTacToe!
Player 1 is X and Player 2 is O
Take turns placing your pieces - the first to 3 in a row wins!
    0   1   2
0   -   -   -
1   -   -   -
2   -   -   -
```

HINTS:

- Blank `print()` statements can be used to create a new line for the board.
- `\t` can be used to create blank space between string values. This can be used to make sure your `-` are the same distance apart from one another. For example, `print("\tWelcome to Tic Tac Toe!")` will print *Welcome to Tic Tac Toe!* tabbed over one space.
- Use a loop to iterate over each row in the board - this will make it easier to print!

B) **Take Turn**

In this exercise, you will add the ability for a user to take a turn by creating four functions:

1. `is_valid_move()` – This function takes a row, col value and returns whether the row,col is a valid move.
2. `place_player()` - Places a player's icon at a given row, col position on the board.
3. `take_manual_turn()` - This function asks the user for a row and a col value until the user provides a valid move. Then, it places that player's icon in the correct spot on the board and prints the board.
4. `take_turn()` - For now, this function should print which player's turn it is and call the `take_manual_turn()` function

The first two functions should be used in the `take_manual_turn()` function.

Here is what a successful implementation of this might look like:

```
Welcome to TicTacToe!
Player 1 is X and Player 2 is O
Take turns placing your pieces - the first to 3 in a row wins!
    0   1   2
0   -   -   -
1   -   -   -
2   -   -   -
X's Turn
Enter a row: 0
Enter a column: 3
Please enter a valid move.
Enter a row: 0
Enter a column: 0
    0   1   2
0   X   -   -
1   -   -   -
2   -   -   -
```

C) **Check Win**

In order to figure out who won the game, we need to create a series of functions to check the board for a winner.

You will create three functions - `check_col_win`, `check_row_win`, and `check_diag_win` that check the board for each possible winning combination, and one function `check_win` that checks to see if one of the three win check functions are true. You should also create a `check_tie` function that returns true if all spaces on the board are taken, and there is no winner.

Here is a brief description of each function:

1. `check_col_win(player)` - returns true if `player` has won in any of the three columns. A player wins if they have three consecutive X's or O's in a column.
2. `check_row_win(player)` - returns true if `player` has won in any of the three row. A player wins if they have three consecutive X's or O's in a row.
3. `check_diag_win(player)` - returns true if `player` has won in either diagonal direction. A player wins if they have three consecutive X's or O's in a diagonal.
4. `check_win(player)` - returns true if `player` has won the game.
5. `check_tie()` - returns true if all spots on the board have been taken, and there is no winner.

D) **Complete the Game**

A game of Tic Tac Toe starts by asking a player to take a turn. Players alternate taking turns until one of the players wins or there are no more spaces left to go on the board, resulting in a tie.

Write the method `play_game()` by creating a while loop that checks after every iteration if someone has won the game, or if the game has resulted in a tie. Remember to use the functions you created previously. You should also make sure to change the value of `player` on each iteration so that the game alternates between players.

When the game is over, your program should indicate which player won, or if the game resulted in a tie.

```
    0   1   2
0   X   -   O
1   X   -   -
2   -   O   -
X's Turn
Enter a row: 2
Enter a column: 0
    0   1   2
0   X   -   O
1   X   -   -
2   X   O   -
X wins!
```