Assignment 3: Making the Game

Objective

In this assignment, you will complete the game interface for a two-player Connect 4 game using a provided game template. This includes creating a user-friendly board display, handling user input, and managing the full game loop including checking for wins, ties, and offering the option to play again.

Instructions

- 1. Complete each function described below in the order provided.
- Use the helper functions you implemented in Assignments 1 and 2 (from the functions module) to manage the board.
- 3. Use the constants provided at the top of the file (NUM_ROWS, NUM_COLS, STREAK_LENGTH, and TOKENS).
- 4. Test your game by running the script and playing it in the console.

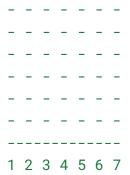
Function Descriptions

display_board(board)

This function is responsible for printing the current state of the board in a user-friendly way.

- Use the TOKENS dictionary to display "X" for player 1, "0" for player 2, and "-" for empty slots. Print column numbers at the bottom to help users choose where to drop their tokens.
- Print each row of the board from top to bottom.

Example Output:



get_drop_column(board, player)

This function asks the current player to choose a column where they want to drop their token.

- Prompt the user for input using input().
- Validate the input:
 - o Is it a digit?
 - o Is it within the valid range of columns (0 to NUM COLS 1)?
 - o Is the column available?
- If the input is invalid, display a helpful error message and re-prompt the player.

Hint: Use functions.column_available() to check whether a column is full.

Return: a valid column index where the token will be dropped.

play()

This function controls one round of the Connect 4 game.

- 1. Use functions.make_board() to create a new game board.
- 2. Alternate turns between player 1 and player 2.
- 3. For each turn:
 - Display the board.
 - Ask the player for their move using get_drop_column().
 - Drop their token into the selected column using functions.drop_token().
 - Check for a win using functions.check_win():
 - If someone wins, display the final board and announce the winner.
 - End the game loop.
 - Check for a tie using functions.board_full():
 - If the board is full and no one has won, display the final board and declare a tie.
 - End the game loop.
- 4. Continue looping until there is a winner or a tie.

```
show_start_screen()
```

You can create a simple introduction screen that runs before the game starts. For example, print:

```
Welcome to Connect 4!

Player 1: X

Player 2: 0

First to connect 4 in a row wins!
```

```
show_end_screen()
```

Display a goodbye message at the end of the game session. For example:

• Thanks for playing Connect 4!

What You'll Have When You're Done

By completing this assignment, you'll have a fully working command-line version of Connect 4 that:

- Displays the game board.
- Accepts valid player moves.
- Handles wins and ties.
- Lets players play again.