coursera

Back to Week 3 | Lessons | This Course: Principles of Computing (Part 2) | Prev | Next

TTTBoard Class

The following is a description of the **TTTBoard** class and its methods as well as several constants associated with this class. Note the code for this class and its constants is imported into the program template for the mini-project via the statement:

```
import poc_ttt_provided as provided

```

You should not need to look at the implementation of this class to complete the project.

Constants for Tic-Tac-Toe

The **TTTBoard** class uses four important constants from the provided code that you will use in your code. These constants are EMPTY, PLAYERX, PLAYERO, and DRAW. You should use these constants in your code when working with methods in the **TTTBoard** class and avoid hard coding any specific values (like 2 or 3) for these constants. Included with the definition of these constants is a helper function **switch_player(player)** that returns PLAYERO on input PLAYERX and PLAYERX on input PLAYERO.

Remember that you will need to add the module name as a prefix when accessing these constants. As the module is imported with the name **provided**, these constants can be referenced as **provided**.EMPTY, **provided**.PLAYERX, **provided**.PLAYERO, **provided**.DRAW. The function **switch_player** can be called as **provided.switch_player(player)**.

Class interface for Tic-Tac-Toe

The **TTTBoard** class has the following interface:

```
class TTTBoard:
    """
    Class to represent a Tic-Tac-Toe board.
    """

    def __init__(self, dim, reverse = False, board = None):
        """
        Initialize the TTTBoard object with the given dimension and
        whether or not the game should be reversed.
        """

    def __str__(self):
        """
        Human readable representation of the board.
        """

    def get_dim(self):
        """
        Return the dimension of the board.
        """

    def square(self, row, col):
        """
        Returns one of the three constants EMPTY, PLAYERX, or PLAYERO
        that correspond to the contents of the board at position (row, col).
        """

    def get_empty_squares(self):
        """
        Return a list of (row, col) tuples for all empty squares
        """

    def move(self, row, col, player):
        """
        Place player on the board at position (row, col).
        player should be either the constant PLAYERX or PLAYERO.
        Does nothing if board square is not empty.
        """

    def check_win(self):
        """
        Returns a constant associated with the state of the game
            If PLAYERX wins, returns PLAYERX.
            If PLAYERO wins, returns PLAYERO.
            If game is drawn, returns DRAW.
            If game is in progress, returns None.
        """

    def clone(self):
        """
        Return a copy of the board.
        """
```

Mark as completed