Connect Four

Weeks 1 and 2

In this laboratory, you are going to develop a textual interface for two people playing Connect Four[1]. The board is a vertically suspended grid of seven columns and six rows.

> The players drop in turn tokens into the columns. The pieces fall straight down, occupying the lowest available space with the column. The objective of the game is to be the first to form a horizontal, vertical, or diagonal line of four of one's own tokens[2].

The grid should be displayed to the player as textual representation with `'.'` representing an empty splot.

```
[['.', '.', '.', '.', '.', '.', '.'],
 ['.', '.', '.', '.', '.', '.', '.'],
 ['.', '.', '.', '.', '.', '.', '.'],
 ['.', '.', '.', '.', '.', '.', '.'],
 ['.', '.', '.', '.', '.', '.', '.'],
 ['.', '.', '.', '.', '.', '.', '.']]
```

The following board shows a field where player 2 (token `'o'`) has won.

```
[['.', '.', '.', '.', '.', '.', '.'],
 ['.', 'x', '.', '.', '.', '.', '.'],
 ['.', 'o', '.', '.', 'x', '.', '.'],
 ['.', 'x', '.', 'o', 'o', '.', '.'],
 ['x', 'x', 'x', 'o', 'o', 'o', 'o'],
 ['x', 'x', 'o', 'x', 'x', 'o', 'o']]
Player 2 wins the game. Congratulations!
```

1 Laboratory tasks

Develop a Python program, which starts with the following header:

```
"""
Textual interface for playing Connect Four with two players.
"""
from pprint import pprint

n_rows_ = 6
n_columns_ = 7
```

[1] https://en.wikipedia.org/wiki/Connect_Four

[2] \{https://en.wikipedia.org/w/index.php?title=Connect_Four&oldid=1073202470

The program should implement the following functions

- `init_field()`,
- `drop_token()`,
- `game_is_won()`, and
- `play()`

as defined below. Additional functions can be defined as one likes. Document your code to an appropriate level, such that another developer can understand and potentially debug your code.

1.1 `init_field()`

```
def init_field(rows=n_rows_, columns=n_columns_):
    """
    Intialize field

    :param rows: number of rows
    :param columns: number of columns
    :return: list of lists
    """
    pass
```

1.2 `drop_token()`

```
def drop_token(field, col, player, symbol={True: 'x', False: 'o'}):
    """
    Modify field with player's choice to drop a token into a specific column

    :param field: Two-dimensional object
    :param col: Column to drop token to
    :param player: Player, whose token is dropped
    :param symbol: dictionary mapping player to token symbol. True refers to player 1,
↪   False to player 2 (not player 1).
    :return: None
    """
    pass
```

1.3 `game_is_won()`

```
def game_is_won(field, void='.'):
    """
    Test if field indicates that one of the players has won the game.

    :param field: Double indexed object
    :return: bool
    """
    pass
```

1.4 `play()`

```
def play(field, tokens):
    """
    Play connect four starting with an intialised field and a corresponding list of
↪   token numbers per column.

    :param field: Object with two indexable dimensions
    :param tokens: List of token numbers per column
    :return: Winner of the game
    """
    pass
```

2 Main

The implemented functions should allow to play the game using the following script:

```
if __name__ == "__main__":
    field, tokens = init_field()
    pprint(field)
    winner = play(field, tokens)
    print(f"Player {winner} wins the game. Congratulations!")
```

3 Submission instructions

Submit the following files as ZIP archive:

- The Python script `connect_four.py`, which contains your implementation of the functions defined above.
- A short one page (DIN A4) report summarizing in bullet points your approach for implementing the functions.
  - Which parts of your code were challenging?
  - Which parts of your code were inspiring?
  - Do you have any suggestions for improving the functions' definitions?