Introduction to Python

ENGSCI233 Lecture 2

Andreas W. Kempa-Liehr

> The Analytical Engine has no pretensions whatever to originate anything. It can do whatever we know how to order it to perform.
>
> Ada Lovelace (Menabrea, 1842, Note G)

1 Programming an textual interface to tic-tac-toe

o		x
o	x	
o	x	

- Simple textual interface for setting tokens onto the 3x3 field.
- Algorithm should be able to detect, if a player has won.
- No *artificial intelligence*.

1.1 Brainstorming

1.2 Implementation

Please watch the Panopto Video `ENGSCI233_PY02c-term1223_tic_tac_toe_coding` for the development process. The developed code is given in the following pages.

```
"""
Textual interface for playing tic tac toe.

No artificial intelligence provided.
"""

def print_hi(name):
    # Use a breakpoint in the code line below to debug your script.
    print(f'Hi, {name}!')    # Press F8 to toggle the breakpoint.

def print_field(field):
    """
    Print the updated field

    :param field: Any printable object
    :return: None
    """
    for i in range(3):
        row = '|'.join(field[i])
        print(row)
        if i < 2:
            print('+'.join(['-'] * 3))


def get_input(field, username):
    """
    Get input from player.

    :param field: Object accepting two concatenated item operators object[][]
    :return: valid row and column tuple
    """
    position = input(f"Player {username}, please enter your move as two comma separated
    ↪  values:")
    idx = [int(x) for x in position.split(',')]
    row, column = idx
    return row, column

def is_won(field, void=' '):
    """
    Check if game has been won and return winning party.

    The overall workflow in the game needs to assure that the player, who made his/her
↪   last move, is known, in order
    to determine the winning player.

    :param field:
    :return: Boolean indicating that the game is won
    """

    for row in field:
        if row[0] != void and row[0] == row[1] == row[2]:
            return True
    for col in range(3):
        if field[col][0] != void and field[col][0] == field[col][1] == field[col][2]:
            return True
    if field[0][0] != void and field[0][0] == field[1][1] == field[2][2]:
        return True
    if field[2][0] != void and field[2][0] == field[1][1] == field[0][2]:
        return True
    return False

def init_field():
```

```
    """
    Return the empty field

    :return: Object accepting two concatenated item operators object[][]
    """
    field = [[' ', ' ', ' '],
             [' ', ' ', ' '],
             [' ', ' ', ' ']]
    return field


def switch_player():
    global active_player
    if active_player == username1:
        active_player = username2
    else:
        active_player = username1


# Press the green button in the gutter to run the script.
if __name__ == '__main__':
    username1 = input("Please enter your name, player 1:")
    username2 = input("Please enter your name, player 2:")
    player = {username1: "o",
              username2: "x"}
    field = init_field()
    print_field(field)

    active_player = username1
    while not is_won(field):
        row, column = get_input(field, active_player)
        field[row][column] = player[active_player]
        print_field(field)
        switch_player()
    switch_player()
    print(f"Congratulations {active_player}. You won the game!")
```

References

L. F. Menabrea. Sketch of the analytical engine invented by Charles Babbage. with notes upon the memoir of the translator Ada Augusta, Countess of Lovelace. *Bibliothéque Universelle de Genéve*, 82, 1842. URL https://www.fourmilab.ch/babbage/sketch.html.