Université de Haute-Alsace
Université franco-azerbaïdjanaise

Master DSAI
2020-2021

Project-Mode applied Programming in Python

Practical Work : Python for Tic-Tac-Toe

Goals

Recall the python basics
Implement Tic-Tac-Toe game
Manage user inputs
Discover object-based python programming

Instructions

Use Python at 100%, Python comes with lot of tools to make your life easy, use them.

Pensez "objet", sachez utiliser les avantages de la programmation objet.

Code neatly with well-chosen variables and functions/methods names. Add useful comments to your code in order to be able to understand it in some days.

Try to respect coding styleguides. I advise you to follow python styleguide PEP8[1] or Google Python Styleguide[2].

Simple is beautiful. Do not try to code complicated, keep it simple, it will be more efficient and less error-prone.

Think before you code, take some time to draw/write your idea on a sheet. The tinking time before you code will save you a lot of debugging time after.

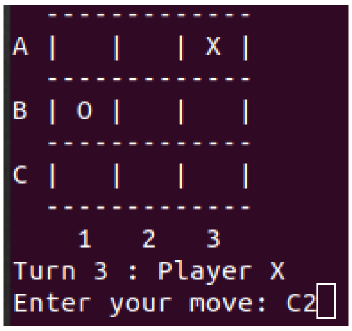
1 Tic-Tac-Toe basic

In this section, we will try to implement a basic version of the well-know Tic-Tac-Toe game[3]. But before going straight to the coding phase, we will think a little bit ...

1. Fix the different steps and states of the game and their evolution
2. Define how you will handle the player decision (moves)
3. Define the data structure you will use to handle the board and to manage the player marks and the turn counter
4. Define the different functions that you will need to handle the game

Now, we have to imagine the interface, for this basic version we will made a console version of Tic-Tac-Toe, an exemple is shown in the snapshot below. In this example, we have a the board with existing marks. We indicate to the user which turn we are and who is the current player. Moreover, we ask the player to enter its move.

1. PEP8 : `https://www.python.org/dev/peps/pep-0008/`
2. Google Python Styleguide : `https://google.github.io/styleguide/pyguide.html`
3. Tic-Tac-Toc on wikipedia : `https://en.wikipedia.org/wiki/Tic-tac-toe`



```
   -------------
A |   |   | X |
   -------------
B | O |   |   |
   -------------
C |   |   |   |
   -------------
    1   2   3
Turn 3 : Player X
Enter your move: C2
```

Based on this simple interface and the different choices you made before, implement your version of Tic-Tac-Toe.

Some useful hints :
— To handle the board as a 2D-array, Numpy [4] can help you.
— Manage the player inputs with `input()` [5]
— Transform player inputs into moves using a dictionnary [6]
— Do not forget to handle illegal moves (e.g. "D4" or "ofhroifherifh") and move on a non-free square.
— Keep it simple, my version has only 80 lines of code (and you could probably do it with less).

Once your Tic-Tac-Toe works, it's time for improvements, check the list below :
— Do you clear screen between each turn ?
— Do you handle moves entered in lower case (e.g. "a3") ?
— Have you succeed to have the player input on the same line as the "Enter your move :" text ?
— Good practice : Does your code have comments ?
— Good practice : Does your code respect PEP8 [7] ?

Everything done ? Congratulations, you now have a basic but functional implementation of Tic-Tac-Toe. But we can do it better ... see you in next section !

2 Tic Tac Toe object-based

Now that you have a basic version, we can now improve the beauty and the power of our code by making it an object version. Python is an object-based programming language. So, we will take our previous code and put it to object, for this follow the instructions below.

1. Create a class TicTacToe which will handle the game
2. Create a class Board which will represents the board and handle it
3. Include your previous code in these two classes and adapt it to this object content

Some useful hints :
— Read the docs, both the official Python documentation [8] and this W3C short tutorial [9]
— At the end do not forget to have some code to create a game object and to launch the game.

It works ? Fine, your code is now more beautiful but what's the interest ? Using object-based allows your code to be reused easily. For example, now, we can easily launch a new game when one is over by simply adding a method to the class TicTacToe reinitializing the game. Test it by modifying your current code without forgetting to ask the players if they want to play a new game.

4. Numpy : `https://numpy.org/`
5. input() : `https://www.w3schools.com/python/ref_func_input.asp`
6. Python dictionnaries : `https://docs.python.org/3/tutorial/datastructures.html#dictionaries`
7. PEP8 : `https://www.python.org/dev/peps/pep-0008/`
8. Python classes : `https://docs.python.org/3/tutorial/classes.html`
9. W3C Python classes tutorial : `https://www.w3schools.com/python/python_classes.asp`

3 Bonus

You already finished ? Congratulations, now you can try to do the same but for the **Connect 4** game [10].

10. Connect 4 on wikipedia :`https://en.wikipedia.org/wiki/Connect_Four`