Tic Tac Toe / 3 en ratlla / 3 en ralla

```
Your move O for x: 2
Your move O for y: 3

     1   2   3
  -------------
1 | O | X | O |
  -------------
2 | X | O | O |
  -------------
3 | X | O | X |
  -------------

Nobody wins in war
```

Carles Solution

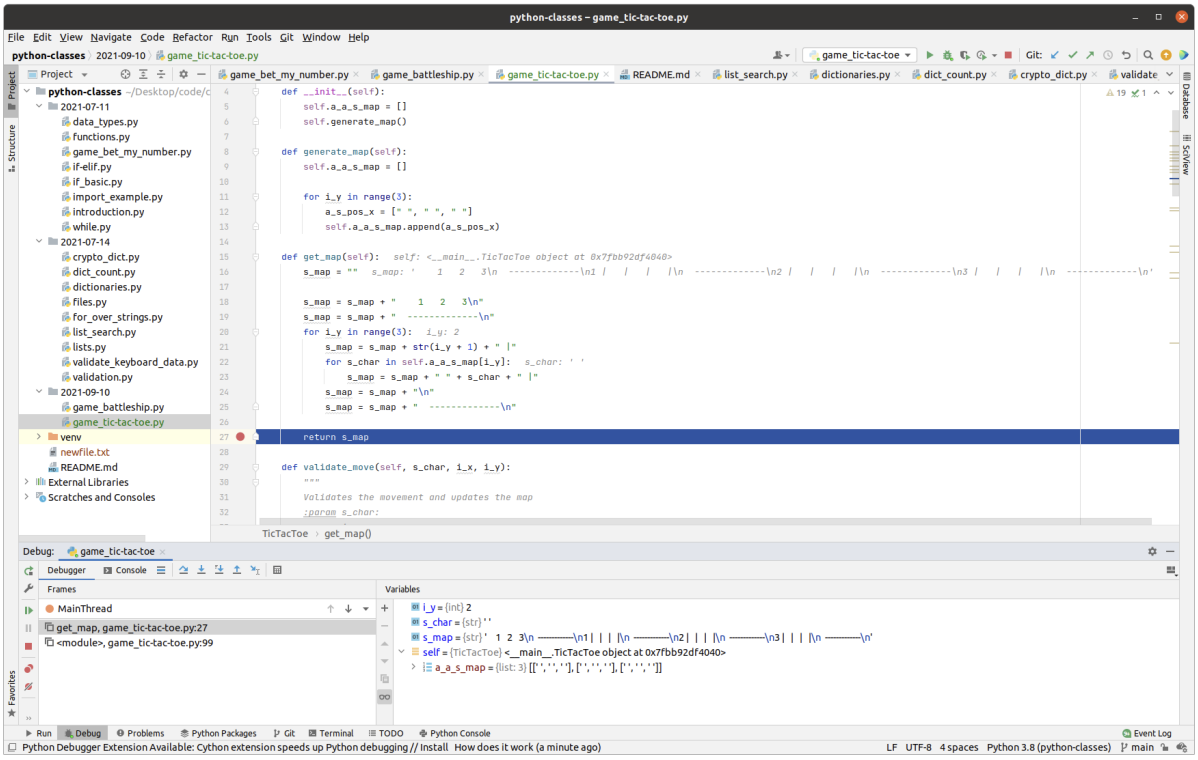
Source Code available here:
https://gitlab.com/carles.mateo/python-classes/-/blob/main/2021-09-10/game_tic-tac-toe.py

```
class TicTacToe:

   def __init__(self):
       self.a_a_s_map = []
       self.generate_map()

   def generate_map(self):
       self.a_a_s_map = []

       for i_y in range(3):
           a_s_pos_x = [" ", " ", " "]
           self.a_a_s_map.append(a_s_pos_x)

   def get_map(self):
       s_map = ""

       s_map = s_map + "    1   2   3\n"
       s_map = s_map + "  -------------\n"
       for i_y in range(3):
           s_map = s_map + str(i_y + 1) + " |"
           for s_char in self.a_a_s_map[i_y]:
               s_map = s_map + " " + s_char + " |"
           s_map = s_map + "\n"
           s_map = s_map + "  -------------\n"

       return s_map
```

```
    def validate_move(self, s_char, i_x, i_y):
        """
        Validates the movement and updates the map
        :param s_char:
        :param i_x:
        :param i_y:
        :return: bool
        """
        i_x = i_x - 1
        i_y = i_y - 1

        if self.a_a_s_map[i_y][i_x] == " ":
            self.a_a_s_map[i_y][i_x] = s_char
            return True

        return False

    def check_win(self):
        for s_char in ["O", "X"]:

            # check horizontal
            for i_y in range(3):
                i_horizontal_match = 0
                for i_x in range(3):
                    if self.a_a_s_map[i_y][i_x] == s_char:
                        i_horizontal_match = i_horizontal_match + 1
                if i_horizontal_match == 3:
                    return True

            # Check vertical
            for i_x in range(3):
                i_vertical_match = 0
                for i_y in range(3):
                    if self.a_a_s_map[i_y][i_x] == s_char:
                        i_vertical_match = i_vertical_match + 1
                if i_vertical_match == 3:
                    return True

            # Check diagonal
            if self.a_a_s_map[1][1] == s_char:
                if self.a_a_s_map[0][0] == s_char and self.a_a_s_map[2][2]
== s_char:
                    return True
                if self.a_a_s_map[0][2] == s_char and self.a_a_s_map[2][0]
== s_char:
                    return True

        return False

    def check_stale(self):
        for i_y in range(3):
            for i_x in range(3):
                if self.a_a_s_map[i_y][i_x] == " ":
                    # Is not full
                    return False

        # We checked all and all were full
        return True
```

```
def get_from_keyboard(s_question, i_min, i_max):
    i_number = 0
    while True:
        s_answer = input(s_question)
        try:
            i_number = int(s_answer)
        except:
            print("Please, type a number")
            continue

        if i_number < i_min or i_number > i_max:
            print("Invalid value. Values should be between", i_min, "and",
i_max)
            continue

        # Validations are Ok
        break

    return i_number


if __name__ == "__main__":
    o_tictactoe = TicTacToe()

    while True:

        s_map = o_tictactoe.get_map()
        print(s_map)

        while True:
            i_x = get_from_keyboard("Your move O for x: ", i_min=1, i_max=3)
            i_y = get_from_keyboard("Your move O for y: ", i_min=1, i_max=3)

            b_valid_move = o_tictactoe.validate_move("O", i_x, i_y)
            if b_valid_move is False:
                print("Invalid move")
                continue

            break

        s_map = o_tictactoe.get_map()
        print(s_map)
        b_check_win = o_tictactoe.check_win()
        if b_check_win is True:
            print("Player O wins!")
            exit(0)

        b_stale = o_tictactoe.check_stale()
        if b_stale is True:
            print("Nobody wins in war")
            exit(0)

        while True:
            i_x = get_from_keyboard("Your move X for x: ", i_min=1, i_max=3)
            i_y = get_from_keyboard("Your move X for y: ", i_min=1, i_max=3)

            b_valid_move = o_tictactoe.validate_move("X", i_x, i_y)
            if b_valid_move is False:
                print("Invalid move")
                continue
```

```
        break

    s_map = o_tictactoe.get_map()
    print(s_map)
    b_check_win = o_tictactoe.check_win()
    if b_check_win is True:
        print("Player X wins!")
        exit(0)
```



Battleship / Enfonsar la flota / Hundir la flota

This is a difficult exercise to do.
The way we face this callenge is by breaking the problem in parts we can manage.
We have already created a PlayerMap Class in the Exercise 3 of Classes section.
That will make it much easier.
So we have to:

- Set how big is the map, and how many ships and which sizes will go
- Position the ships for the Computer
    - Make sure they fit in the map
    - Make sure they have at least one water position between ship's cells
- Position the ships for the Player, or ask them to position manually.
    - Make sure they fit in the map
    - Make sure they have at least one water position between ship's cells
- Display the Computer's map empty (ships hidden)
- Ask the user for a coordinate for their shoot
    - Detect if the shoot impacted on a ship or in the water
- Make the computer shoot, with more or less intelligence
    - Detect if the shoot impacted on a ship or in the water
- Detect if somebody sunk all the ships of the opponent, so they win

In the solution I will explain how I did certain parts and how I could have done it better.

Aquest és un exercici difícil de fer.
La manera d'escometre el repte és trencant el problema en parts que podem manegar.
Ja hem creat una classe PlayerMap a l'exercici 3 de la secció Classes.
Això ho farà molt més fàcil.
Així doncs hem de:

- Fixar com de gran és el mapa, i quants vaixells i de quina mida hi aniran
- Posicionar els vaixells per a l'ordinador
    - Assegurar-nos que són dins el mapa
    - Assegurar-nos que hi ha com a mínim una separació (aigua) entre les cel·les dels diferents vaixells
- Posicionar els vaixells per al jugador, o permetre-li posicionar-los manualment
    - Assegurar-nos que són dins el mapa
    - Assegurar-nos que hi ha com a mínim una separació (aigua) entre les cel·les dels diferents vaixells
- Mostrar el mapa de l'ordinador vuit (el vaixells amagats)
- Demanar a l'usuari les coordinades del seu tret
    - Detectar si el tret ha impactat a un vaixell o a l'aigua
- Fer que l'ordinador dispari, amb més o menys intel·ligència
    - Detectar si el tret ha impactat a un vaixell o a l'aigua
- Detectar si algú ha enfonsat tots els vaixells de l'oponen, i per tant ha guanyat

A la solució explicaré com he solventat certes part i com les podria haver fet millor.

Este es un ejercicio difícil de hacer.
La manera de acometer el reto es rompiendo el problema en partes que podemos manejar.
Ya hemos creado una clase PlayerMap en el ejercicio 3 de la sección Classes.
Esto lo hará mucho más fácil.
Así pues hemos de:

- Fijar como de grande es el mapa, y cuantos barcos y de qué medida serán
- Posicionar los barcos para el ordenador
    - Asegurarnos que están dentro de los márgenes del mapa
    - Asegurarnos que hay como mínimo una separación (agua) entre las celdas de los diferentes barcos
- Posicionar los barcos para el jugador, o permitirle posicionarlos manualmente
    - Asegurarnos que están dentro de los márgenes del mapa
    - Asegurarnos que hay como mínimo una separación (agua) entre las celdas de los diferentes barcos
- Mostrar el mapa del ordenador vacío (los barcos ocultos)
- Solicitar al usuario las coordenadas de su disparo
    - Detectar si el disparo ha impactado en un barco o en el mar
- Hacer que el ordenador dispare, con más o menos inteligencia
    - Detectar si el disparo ha impactado en un barco o en el mar
- Detectar si alguien h hundido todos los barcos del oponente, y por tanto ha ganado

En la solución explicaré cómo he solventado ciertas partes y cómo las podría haber solucionado mejor.

Me in the middle of a game:

Jo enmig del joc:

Yo en medio del juego:

```
Choose your coordinates to fire (i.e.: A1) :P19
Water!
The computer fired on C8
The computer hit you!
Player's map:
    A B C D E F G H I J K L M N O P Q R S T
   ____________________________________________
 1 |*|X|*| | |.| | |.| | | | | |*|*|*|.|.|*|
 2 |*|3|*| | | | | | |.| | | |*|4|4|4|4|*|
 3 |*|3|.| | |.| |.| |.| |.|.|.|*|*|.|.|*|*|
 4 |*|*|.|*|*|*|*|*|.| |.| | | | | |.| | | |
 5 |*|.|*|*|4|4|4|4|*|.|.| |.| | | | | |.| |
 6 |.|7|*|*|*|*|*|*|*|.| |*|*|*|.| | | |.| |
 7 |*|7|*| | | | | |.| |.|*|5|*| | |*|*|*|.|
 8 |*|X|*|*|*|*|*|*| | | |.|5|*| |.|*|4|*|.|
 9 |*|7|*|*|X|3|3|*|.|.|*|.|5|*|.| |.|4|*| |
10 |.|7|*|*|*|*|.|.|*|3|*|*|X|*| | |*|4|*| |
11 |*|7|*| | | |.| |.|X|*|*|5|*|.| |*|4|*| |
12 |*|7|*| | | | | |*|3|*|*|*|*| | |*|.|.| |
13 |.|*|*| | | | | |*|*|*|.| | |.| | | | | |
14 | |.| |.|.|*| | | | | | |.|.| | |*|*|*|.|
15 | |.| |*|X|.| | | | |.| | | |.| |*|3|3|3|
16 | | | |.|5|*| | |.| | | | | | | |*|*|.|.|
17 | | | |*|X|*| | | | |.| | |.|*|*| | | | |
18 | | |.|*|5|*| | | | | |.|.|*|3|*|.| | | |
19 | | | |.|5|*| | |.| |.| | |*|3|*| | | | |
20 | |.| |.|.|*| |.| | | | | |.|3|*| | | | |
   ----------------------------------------

Computer's map:
    A B C D E F G H I J K L M N O P Q R S T
   ____________________________________________
 1 |.|?|?|?|?|.|?|?|?|.|?|?|?|.|.|?|?|?|?|.|
 2 |?|.|.|.|?|.|?|.|?|?|.|.|.|X|X|X|X|X|.|?|?|
 3 |?|?|?|?|?|?|?|?|?|?|?|?|?|?|.|?|?|?|?|?|?|
 4 |?|?|.|?|.|?|?|?|?|?|.|?|?|?|.|?|?|.|?|?|?|
 5 |.|?|?|?|X|?|.|?|?|?|?|?|?|?|?|?|?|?|?|.|
 6 |?|?|?|?|X|?|?|?|?|?|?|.|?|?|?|.|?|?|?|.|
 7 |.|?|?|?|X|.|?|?|?|?|?|?|?|?|.|?|?|?|?|?|?|
 8 |?|?|?|?|X|?|?|?|?|?|?|.|?|?|.|?|?|?|?|?|?|
 9 |?|?|?|?|X|?|?|?|?|?|?|?|?|?|X|?|?|.|?|?|?|
10 |.|?|?|?|X|?|?|?|?|.|?|?|?|?|X|.|.|X|?|?|.|
11 |?|?|?|?|X|?|?|?|?|?|?|?|?|?|X|?|?|X|?|?|?|
12 |.|?|?|?|.|?|?|?|?|?|.|?|?|.|?|?|X|?|?|?|
13 |?|?|?|?|?|?|?|?|?|?|?|?|?|?|.|?|?|X|?|?|?|
14 |?|?|?|?|?|?|.|?|?|?|?|?|?|?|X|?|?|.|?|?|?|
15 |?|?|?|.|?|?|?|?|.|?|?|?|?|?|X|?|?|?|?|?|?|
16 |?|?|?|.|?|?|?|.|.|?|?|?|?|?|X|?|.|?|?|?|?|
17 |.|?|?|?|?|?|?|?|?|?|?|?|?|?|.|?|?|?|.|?|?|
18 |?|.|?|?|?|.|X|X|X|X|.|?|?|?|?|.|?|?|.|?|
19 |?|?|.|?|?|?|.|?|?|.|?|?|?|?|?|.|?|?|?|?|
20 |?|?|?|.|?|?|?|?|?|?|.|?|?|?|?|?|.|?|?|?|
   ----------------------------------------
```