THE BIG STEPS TO A FIRST WORKING GAME :

* CREATE THE BOARD
* CREATE THE SHIPS AND PUT THEM ON THE BOARD
* MANAGE GAME FLOW

PATCH NOTES:

V0.2: Added 3rd and 4th value to board’s coordinates : H for HIT and M for MISSED

V0.3: Added function that returns a visual progress of the game

V0.4: Function that returns a visual progress of the game

V0.5: Recoded boat placement + extended board + more explanations

V0.6: Board display + error on shoot try if position unavailable + small tweaks

V0.7: Added second plain\_board

V0.8: Inverted board display + small prompt tweaks

TO IMPLEMENT :

Turn by turn Player 2  
Change size of ships from 1 to x =< 5 horizontal

Implement different ships

Manage ship orientation

Automate board creation instead of feeding it

Misc:  
Comprehensive error of input management at all stages

I. Représentation visuelle de l’échiquier:

'a1','a2','a3','a4','a5',

'b1','b2','b3','b4','b5',

'c1','c2','c3','c4','c5',

'd1','d2','d3','d4','d5',

'e1','e2','e3','e4','e5',

II. first code draft

board =

['a1','a2','a3','a4','a5',

'b1','b2','b3','b4','b5',

'c1','c2','c3','c4','c5',

'd1','d2','d3','d4','d5',

'e1','e2','e3','e4','e5']

board = {‘a1’: ‘o’, ‘a2: x',’'a3':’’o’,'a4’:’o', 'a5': o, 'b1: o', 'b2': o,'b3': ‘o’,'b4': ‘o’, 'b5': ‘o’, 'c1': ‘o’, 'c2': ‘o’, 'c3': ‘x’,'c4’: ‘o’,'c5’:’o’,'d1’o’,'d2’:’o’, 'd3’:’o’,'d4’:’o’','d5’:‘o', 'e1’:’o’,'e2':’o’,'e3':’o’, 'e4': ‘x’, 'e5': ‘o’}

board = {

"A1": "o",

"A2": "o",

"A3": "o",

"A4": "o",

"A5": "o",

"B1": "o",

"B2": "o",

"B3": "o",

"B4": "o",

"B5": "o",

"C1": "o",

"C2": "o",

"C3": "o",

"C4": "o",

"C5": "o",

"D1": "o",

"D2": "o",

"D3": "o",

"D4": "o",

"D5": "o",

"E1": "o",

"E2": "o",

"E3": "o",

"E4": "o",

"E5": "o",

}

Code : choix position

key = input("Please enter a key for the position: ")

for i in board:

board[key] = 'x'

print(board)

Code : condition de victoire et boucles correspondantes:

win\_condition == False

while win\_condition == False:

for board\_checker in board:

if board.values == ‘o’:

win\_condition = True

print('It’s a win!')

break

else:

input("Pick another position to shoot on: ")

key = input("Please chose where you want to place your ship")

for i in board:

board[key] = 'x'

for board\_checker in board:

if board.values == ‘o’:

win\_condition = True

print('It’s a win!')

break

CODE ROBIN :

# Online Python compiler (interpreter) to run Python online.

# Write Python 3 code in this online editor and run it.

# Online Python compiler (interpreter) to run Python online.

# Write Python 3 code in this online editor and run it.

# Online Python compiler (interpreter) to run Python online.

# Write Python 3 code in this online editor and run it.

# Online Python compiler (interpreter) to run Python online.

# Write Python 3 code in this online editor and run it.

board = {

"A1": "o",

"A2": "o",

"A3": "o",

"A4": "o",

"A5": "o",

"B1": "o",

"B2": "o",

"B3": "o",

"B4": "o",

"B5": "o",

"C1": "o",

"C2": "o",

"C3": "o",

"C4": "o",

"C5": "o",

"D1": "o",

"D2": "o",

"D3": "o",

"D4": "o",

"D5": "o",

"E1": "o",

"E2": "o",

"E3": "o",

"E4": "o",

"E5": "o",

}

# Online Python compiler (interpreter) to run Python online.

# Write Python 3 code in this online editor and run it.

# Online Python compiler (interpreter) to run Python online.

# Write Python 3 code in this online editor and run it.

board = {

"A1": "o",

"A2": "o",

"A3": "o",

"A4": "o",

"A5": "o",

"B1": "o",

"B2": "o",

"B3": "o",

"B4": "o",

"B5": "o",

"C1": "o",

"C2": "o",

"C3": "o",

"C4": "o",

"C5": "o",

"D1": "o",

"D2": "o",

"D3": "o",

"D4": "o",

"D5": "o",

"E1": "o",

"E2": "o",

"E3": "o",

"E4": "o",

"E5": "o",

}

ship\_position = input("Please enter a position to place your ship: ")

for i in board:

board[ship\_position] = 'x'

print(board)

first\_turn\_board\_list = list(board.values())

print(first\_turn\_board\_list)

first\_turn\_shoot = (input("Pick a first position to fire at: "))

win\_condition = False

while win\_condition == False:

if "x" not in first\_turn\_board\_list:

print("It’s a win! Congratz!")

break

else:

new\_turn\_shoot = (input("Pick a new position to fire at: "))

if new\_turn\_shoot in board:

if board[new\_turn\_shoot] == "o":

board.update({new\_turn\_shoot:"x"})

new\_turn\_board\_list = list(board.values())

print(new\_turn\_board\_list)

# if board\_list == ():

# print(board)

board = {

"A1": "o",

"A2": "o",

"A3": "o",

"A4": "o",

"A5": "o",

"B1": "o",

"B2": "o",

"B3": "o",

"B4": "o",

"B5": "o",

"C1": "o",

"C2": "o",

"C3": "o",

"C4": "o",

"C5": "o",

"D1": "o",

"D2": "o",

"D3": "o",

"D4": "o",

"D5": "o",

"E1": "o",

"E2": "o",

"E3": "o",

"E4": "o",

"E5": "o",

}

ship\_position = input("Please enter a position to place your ship: ")

for i in board:

board[ship\_position] = 'x'

print(board)

first\_turn\_board\_list = list(board.values())

print(first\_turn\_board\_list)

first\_turn\_shoot = (input("Pick a first position to fire at: "))

win\_condition = False

while win\_condition == False:

if "x" not in first\_turn\_board\_list:

print("It’s a win! Congratz!")

break

else:

new\_turn\_shoot = (input("Pick a new position to fire at: "))

if new\_turn\_shoot in board:

if board[new\_turn\_shoot] == "x":

board.update({new\_turn\_shoot:"o"})

new\_turn\_board\_list = list(board.values())

print(new\_turn\_board\_list)

**Andy feedback:**

Andy told us we should create two lists.

Dreading the idea of starting over from scratch, we decided to stick to our dictionary.

DRAFT:

board = {

"A1": "o",

"A2": "o",

"A3": "o",

"A4": "o",

"A5": "o",

"B1": "o",

"B2": "o",

"B3": "o",

"B4": "o",

"B5": "o",

"C1": "o",

"C2": "o",

"C3": "o",

"C4": "o",

"C5": "o",

"D1": "o",

"D2": "o",

"D3": "o",

"D4": "o",

"D5": "o",

"E1": "o",

"E2": "o",

"E3": "o",

"E4": "o",

"E5": "o",

}

ship\_position = input("Please enter a position to place your ship: ")

for i in board:

board[ship\_position] = 'x'

print(board)

win\_condition = False

# board\_checker = 0

turn\_shoot = (input("Pick a first position to fire at: "))

while win\_condition == False:

turn\_board\_list = list(board.values())

print(turn\_board\_list)

for board\_checker in turn\_board\_list:

# board\_checker = int(board\_checker)

if turn\_board\_list[0:] != "x" and win\_condition == False:

win\_condition = True

print('It’s a win! Congratz!')

break

else:

input("Pick a position to fire at: ")

board\_list

print(board)

17h09

ship\_position = input("Please enter a position to place your ship: ")

for i in board:

board[ship\_position] = 'x'

print(board)

first\_turn\_board\_list = list(board.values())

print(first\_turn\_board\_list)

first\_turn\_shoot = (input("Pick a first position to fire at: "))

win\_condition = False

while win\_condition == False:

if "x" not in first\_turn\_board\_list:

print("It’s a win! Congratz!")

break

else:

new\_turn\_shoot = (input("Pick a new position to fire at: "))

if new\_turn\_shoot in board: #key

if board[new\_turn\_shoot] == "x": #value

board.update({new\_turn\_shoot:"o"}) #transforme value en "o" no matter what was value

new\_turn\_board\_list = list(board.values())

print(new\_turn\_board\_list)

17h13 - CA MARCHE

ship\_position = input("Please enter a position to place your ship: ")

for i in board:

board[ship\_position] = 'x'

print(board)

turn\_board\_list = list(board.values())

print(turn\_board\_list)

first\_turn\_shoot = (input("Pick a first position to fire at: "))

win\_condition = False

while win\_condition == False:

if "x" not in turn\_board\_list:

print("It’s a win! Congratz!")

break

else:

new\_turn\_shoot = (input("Pick a new position to fire at: "))

if new\_turn\_shoot in board: #key

if board[new\_turn\_shoot] == "x": #value

board.update({new\_turn\_shoot:"o"}) #transforme value en "o" no matter what was value

turn\_board\_list = list(board.values())

print(turn\_board\_list)

17h25 - PLUS SUR QUE CA MARCHE

ship\_position = input("Please enter a position to place your ship: ")

for i in board:

board[ship\_position] = 'x'

print(board)

turn\_board\_list = list(board.values())

print(turn\_board\_list)

first\_turn\_shoot = (input("Pick a first position to fire at: "))

win\_condition = False

while win\_condition == False:

if "x" not in turn\_board\_list:

print("It’s a win! Congratz!")

break

else:

new\_turn\_shoot = (input("Pick a new position to fire at: "))

if new\_turn\_shoot in board: #key

if board[new\_turn\_shoot] == "x": #value = bateau

board.update({new\_turn\_shoot:"H"}) #transforme value en "H" car touché "H"

elif board[new\_turn\_shoot] == "o": #value = case vide

board.update({new\_turn\_shoot:"M"}) #transforme value en "M" car plouf dans l'eau

else:

board.update({new\_turn\_shoot:"T"}) #transforme value en "T" car tried

print("Vous avez déjà essayé cette position.")

turn\_board\_list = list(board.values())

print(turn\_board\_list)

17h35 - CA MARCHOUILLE MAIS ON PEUT MIEUX FAIRE - DOUBLON NECESS. POUR WIN

ship\_position = input("Please enter a position to place your ship: ")

for i in board:

board[ship\_position] = 'x'

print(board)

turn\_board\_list = list(board.values())

print(turn\_board\_list)

# new\_turn\_shoot = (input("Pick a position to fire at: "))

win\_condition = False

while win\_condition == False:

if "x" not in turn\_board\_list:

print("It’s a win! Congratz!")

break

else:

new\_turn\_shoot = (input("Pick a position to fire at: "))

if new\_turn\_shoot in board: #key

if board[new\_turn\_shoot] == "x": #value = bateau

board.update({new\_turn\_shoot:"H"}) #transforme value en "H" car hit

elif board[new\_turn\_shoot] == "o": #value = case vide

board.update({new\_turn\_shoot:"M"}) #transforme value en "M" car miss

else:

board.update({new\_turn\_shoot:"T"}) #transforme value en "T" car tried

print("Vous avez déjà essayé cette position.")

turn\_board\_list = list(board.values())

print(turn\_board\_list)

17h38: Marchouille mieux

board = {

"A1": "o",

"A2": "o",

"A3": "o",

"A4": "o",

"A5": "o",

"B1": "o",

"B2": "o",

"B3": "o",

"B4": "o",

"B5": "o",

"C1": "o",

"C2": "o",

"C3": "o",

"C4": "o",

"C5": "o",

"D1": "o",

"D2": "o",

"D3": "o",

"D4": "o",

"D5": "o",

"E1": "o",

"E2": "o",

"E3": "o",

"E4": "o",

"E5": "o",

}

ship\_position = input("Please enter a position to place your ship: ")

for i in board:

board[ship\_position] = 'x'

print(board)

turn\_board\_list = list(board.values())

print(turn\_board\_list)

# new\_turn\_shoot = (input("Pick a position to fire at: "))

win\_condition = False

while win\_condition == False:

if "x" not in turn\_board\_list:

print("It’s a win! Congratz!")

break

else:

new\_turn\_shoot = (input("Pick a position to fire at: "))

if new\_turn\_shoot in board: #key

if board[new\_turn\_shoot] == "x": #value = bateau

board.update({new\_turn\_shoot:"H"}) #transforme value en "H" car hit

elif board[new\_turn\_shoot] == "o": #value = case vide

board.update({new\_turn\_shoot:"M"}) #transforme value en "M" car miss

else:

board.update({new\_turn\_shoot:"T"}) #transforme value en "T" car tried

print("Vous avez déjà essayé cette position.")

turn\_board\_list = list(board.values())

print(turn\_board\_list)

17h46 - Enlevé le ‘T’ en cas d’essai précédent sur la même case

ship\_position = input("Please enter a position to place your ship: ")

for i in board:

board[ship\_position] = 'x'

print(board)

turn\_board\_list = list(board.values())

print(turn\_board\_list)

# new\_turn\_shoot = (input("Pick a position to fire at: "))

win\_condition = False

while win\_condition == False:

if "x" not in turn\_board\_list:

print("It’s a win! Congratz!")

break

else:

new\_turn\_shoot = (input("Pick a position to fire at: "))

if new\_turn\_shoot in board: #key

if board[new\_turn\_shoot] == "x": #value = bateau

board.update({new\_turn\_shoot:"H"}) #transforme value en "H" car hit

elif board[new\_turn\_shoot] == "o": #value = case vide

board.update({new\_turn\_shoot:"M"}) #transforme value en "M" car miss

else:

# board.update({new\_turn\_shoot:"T"}) #transforme value en "T" car tried

print("Vous avez déjà essayé cette position.")

turn\_board\_list = list(board.values())

print(turn\_board\_list)

17h55 - 5 bateaux de 1 case à placer

#Placer bateau 1

ship\_position1 = input("Please enter a position to place your ship: ")

for i in board:

board[ship\_position1] = 'x'

#Placer bateau 2

ship\_position2 = input("Please enter a position to place your second ship: ")

for i in board:

board[ship\_position2] = 'x'

#Placer bateau 3

ship\_position3 = input("Please enter a position to place your third ship: ")

for i in board:

board[ship\_position3] = 'x'

#Placer bateau 4

ship\_position4 = input("Please enter a position to place your forth ship: ")

for i in board:

board[ship\_position4] = 'x'

#Placer bateau 5

ship\_position5 = input("Please enter a position to place your fifth ship: ")

for i in board:

board[ship\_position5] = 'x'

print(board) #Affichage du plateau

turn\_board\_list = list(board.values())

print(turn\_board\_list) #Affichage de la liste

win\_condition = False

while win\_condition == False:

if "x" not in turn\_board\_list:

print("It’s a win! Congratz!")

break

else:

new\_turn\_shoot = (input("Pick a position to fire at: "))

if new\_turn\_shoot in board: #key

if board[new\_turn\_shoot] == "x": #value = bateau

board.update({new\_turn\_shoot:"H"}) #transforme value en "H" car hit

elif board[new\_turn\_shoot] == "o": #value = case vide

board.update({new\_turn\_shoot:"M"}) #transforme value en "M" car miss

else:

# board.update({new\_turn\_shoot:"T"}) #transforme value en "T" car tried

print("Vous avez déjà essayé cette position.")

turn\_board\_list = list(board.values())

print(turn\_board\_list)

-New board =

**Andy feedback**

He helped us create a function which would display the gameboard to the user at every round (in a more user friendly way)

18h20 - Création du tableau

def display\_gameboard(record\_list):

start\_slice = 0

end\_slice = 5

for i in range(0,5):

print(record\_list[start\_slice:end\_slice])

start\_slice+=5

end\_slice+=5

record\_list = ['H', 'H', 'o', 'o', 'o', 'H', 'o', 'o', 'o', 'o', 'H', 'o', 'o', 'M', 'o', 'H', 'o', 'o', 'o', 'o', 'o', 'o', 'M', 'o', 'o']

display\_gameboard(record\_list)

18h25: Working board with more feedback and clearer typo

board = {

"A1": "O",

"A2": "O",

"A3": "O",

"A4": "O",

"A5": "O",

"B1": "O",

"B2": "O",

"B3": "O",

"B4": "O",

"B5": "O",

"C1": "O",

"C2": "O",

"C3": "O",

"C4": "O",

"C5": "O",

"D1": "O",

"D2": "O",

"D3": "O",

"D4": "O",

"D5": "O",

"E1": "O",

"E2": "O",

"E3": "O",

"E4": "O",

"E5": "O",

}

#Placer bateau 1

ship\_position1 = input("Please enter a position to place your ship: ")

for i in board:

board[ship\_position1] = 'X'

#Placer bateau 2

ship\_position2 = input("Please enter a position to place your second ship: ")

for i in board:

board[ship\_position2] = 'X'

#Placer bateau 3

ship\_position3 = input("Please enter a position to place your third ship: ")

for i in board:

board[ship\_position3] = 'X'

#Placer bateau 4

ship\_position4 = input("Please enter a position to place your fourth ship: ")

for i in board:

board[ship\_position4] = 'X'

#Placer bateau 5

ship\_position5 = input("Please enter a position to place your five ship: ")

for i in board:

board[ship\_position5] = 'X'

print(board) #Affichage du plateau

turn\_board\_list = list(board.values())

print(turn\_board\_list) #Affichage de la liste

win\_condition = False

while win\_condition == False:

if "X" not in turn\_board\_list:

print("It’s a win! Congratz!")

break

else:

new\_turn\_shoot = (input("Pick a position to fire at: "))

if new\_turn\_shoot in board: #key

if board[new\_turn\_shoot] == "X": #value = bateau

board.update({new\_turn\_shoot:"H"}) #transforme value en "H" car hit

print("\nIt's a HIT! :) ")

elif board[new\_turn\_shoot] == "O": #value = case vide

board.update({new\_turn\_shoot:"M"}) #transforme value en "M" car miss

print("\nIt's a MISS! :( ")

else:

print("\nWATCH OUT! You've already tried this position!!!")

turn\_board\_list = list(board.values())

print("Your progress so far:\n", turn\_board\_list)

{

"A1": "O",

"A2": "O",

"A3": "O",

"A4": "O",

"A5": "O",

"A6": "O",

"A7": "O",

"A8": "O",

"A9": "O",

"A10": "O",

"B1": "O",

"B2": "O",

"B3": "O",

"B4": "O",

"B5": "O",

"B6": "O",

"B7": "O",

"B8": "O",

"B9": "O",

"B10": "O",

"C1": "O",

"C2": "O",

"C3": "O",

"C4": "O",

"C5": "O",

"C6": "O",

"C7": "O",

"C8": "O",

"C9": "O",

"C10": "O",

"D1": "O",

"D2": "O",

"D3": "O",

"D4": "O",

"D5": "O",

"D6": "O",

"D7": "O",

"D8": "O",

"D9": "O",

"D10": "O",

"E1": "O",

"E2": "O",

"E3": "O",

"E4": "O",

"E5": "O",

"E6": "O",

"E7": "O",

"E8": "O",

"E9": "O",

"E10": "O",

"F1": "O",

"F2": "O",

"F3": "O",

"F4": "O",

"F5": "O",

"F6": "O",

"F7": "O",

"F8": "O",

"F9": "O",

"F10": "O",

"G1": "O",

"G2": "O",

"G3": "O",

"G4": "O",

"G5": "O",

"G6": "O",

"G7": "O",

"G8": "O",

"G9": "O",

"G10": "O",

"H1": "O",

"H2": "O",

"H3": "O",

"H4": "O",

"H5": "O",

"H6": "O",

"H7": "O",

"H8": "O",

"H9": "O",

"H10": "O",

"I1": "O",

"I2": "O",

"I3": "O",

"I4": "O",

"I5": "O",

"I6": "O",

"I7": "O",

"I8": "O",

"I9": "O",

"I10": "O",

"J1": "O",

"J2": "O",

"J3": "O",

SAMEDI 10h23  
-Updated a few typos

-Updated commentary

# Online Python compiler (interpreter) to run Python online.

# Write Python 3 code in this online editor and run it.

# Online Python compiler (interpreter) to run Python online.

# Write Python 3 code in this online editor and run it.

def display\_gameboard(record\_list):

start\_slice = 0

end\_slice = 5

for i in range(0,5):

print(record\_list[start\_slice:end\_slice])

start\_slice += 5

end\_slice += 5

board = {

"A1": "O",

"A2": "O",

"A3": "O",

"A4": "O",

"A5": "O",

"B1": "O",

"B2": "O",

"B3": "O",

"B4": "O",

"B5": "O",

"C1": "O",

"C2": "O",

"C3": "O",

"C4": "O",

"C5": "O",

"D1": "O",

"D2": "O",

"D3": "O",

"D4": "O",

"D5": "O",

"E1": "O",

"E2": "O",

"E3": "O",

"E4": "O",

"E5": "O",

}

#Placer bateau 1

ship\_position1 = input("Please enter a position to place your ship: ")

for i in board:

board[ship\_position1] = 'X'

#Placer bateau 2

ship\_position2 = input("Please enter a position to place your second ship: ")

for i in board:

board[ship\_position2] = 'X'

#Placer bateau 3

ship\_position3 = input("Please enter a position to place your third ship: ")

for i in board:

board[ship\_position3] = 'X'

#Placer bateau 4

ship\_position4 = input("Please enter a position to place your fourth ship: ")

for i in board:

board[ship\_position4] = 'X'

#Placer bateau 5

ship\_position5 = input("Please enter a position to place your five ship: ")

for i in board:

board[ship\_position5] = 'X'

print(board) #affichage du plateau

turn\_board\_list = list(board.values())

print(turn\_board\_list) #affichage de la liste

win\_condition = False #création d'une condition pour lancer la partie

while win\_condition == False: #lancement de la boucle de jeu

if "X" not in turn\_board\_list: #condition selon laquelle la partie s'arrête

print("\nIt’s a win! Congratz!") #plus aucun bateau = game over

break

else:

new\_turn\_shoot = (input("Pick a position to fire at: ")) #si la partie n'est pas finie : demande de coordonnées pour le tir

if new\_turn\_shoot in board: #accès à la clé correspondante aux coordonnées indiquées par le joeur

if board[new\_turn\_shoot] == "X": #si valeur associée à la clé = bateau

board.update({new\_turn\_shoot:"H"}) #transforme valeur en "H" pour indiquer "HIT"

print("\nIt's a HIT! :) ")

elif board[new\_turn\_shoot] == "O": #si valeur = case vide

board.update({new\_turn\_shoot:"M"}) #transforme value en "M" pour indiquer "MISS"

print("\nIt's a MISS! :( ")

else:

print("\nWATCH OUT! You've already tried this position!!!") #si valeur n'est ni bateau, ni vide elle a forcément été essayée auparavant

turn\_board\_list = list(board.values()) #mise à jour du board après le tour

print("Your progress so far:\n", turn\_board\_list) #affichage du board pour le joueur

Dimanche 10h33 :

v0.5 - Recoded boat placement + extended board + more explanations

def display\_gameboard(record\_list):

start\_slice = 0

end\_slice = 5

for i in range(0,5):

print(record\_list[start\_slice:end\_slice])

start\_slice += 5

end\_slice += 5

board = {

"A1": "O","A2": "O","A3": "O","A4": "O","A5": "O","A6": "O","A7": "O","A8": "O","A9": "O","A10": "O",

"B1": "O","B2": "O","B3": "O","B4": "O","B5": "O","B6": "O","B7": "O","B8": "O","B9": "O","B10": "O",

"C1": "O","C2": "O","C3": "O","C4": "O","C5": "O","C6": "O","C7": "O","C8": "O","C9": "O","C10": "O",

"D1": "O","D2": "O","D3": "O","D4": "O","D5": "O","D6": "O","D7": "O","D8": "O","D9": "O","D10": "O",

"E1": "O","E2": "O","E3": "O","E4": "O","E5": "O","E6": "O","E7": "O","E8": "O","E9": "O","E10": "O",

"F1": "O","F2": "O","F3": "O","F4": "O","F5": "O","F6": "O","F7": "O","F8": "O","F9": "O","F10": "O",

"G1": "O","G2": "O","G3": "O","G4": "O","G5": "O","G6": "O","G7": "O","G8": "O","G9": "O","G10": "O",

"H1": "O","H2": "O","H3": "O","H4": "O","H5": "O","H6": "O","H7": "O","H8": "O","H9": "O","H10": "O",

"I1": "O","I2": "O","I3": "O","I4": "O","I5": "O","I6": "O","I7": "O","I8": "O","I9": "O","I10": "O",

"J1": "O","J2": "O","J3": "O","J4": "O","J5": "O","J6": "O","J7": "O","J8": "O","J9": "O","J10": "O",

}

#Placer bateau 1

ship\_position1 = input("Please select a position to place your first ship: ")

first\_boat\_placed = False #bateau 1 n'est pas placé

while first\_boat\_placed == False: #tant que bateau 1 n'est pas placé

if ship\_position1 in board: #si position indiquée est bien valide et présente sur le plateau

if board[ship\_position1] != 'X': #si la position n'est pas déjà occupée par un bateau

board[ship\_position1] = 'X' #position indiquée = bateau 1

first\_boat\_placed = True #bateau 1 placé => sortie de boucle de placement du bateau

else:

ship\_position1 = input("Position seems to be occupied by another boat.\nPlease select a new position for your first ship: ")

else:

ship\_position1 = input("The position you provided is not in the game board and therefore invalid.\nPlease select a new valid position for your first ship: ")

# #Placer bateau 2

ship\_position2 = input("Please select a position to place your second ship: ")

second\_boat\_placed = False #bateau 2 n'est pas placé

while second\_boat\_placed == False: #tant que bateau 2 n'est pas placé

if ship\_position2 in board: #si position indiquée est bien valide et présente sur le plateau

if board[ship\_position2] != 'X': #si la position n'est pas déjà occupée par un bateau

board[ship\_position2] = 'X' #position indiquée = bateau

second\_boat\_placed = True #bateau placé => sortie de boucle de placement du bateau

else:

ship\_position2 = input("Position seems to be occupied by another boat.\nPlease select a new position for your first ship: ")

else:

ship\_position2 = input("The position you provided is not in the game board and therefore invalid.\nPlease select a new valid position for your first ship: ")

# #Placer bateau 3

# ship\_position3 = input("Please enter a position to place your third ship: ")

# for i in board:

# if board[ship\_position]:

# board[ship\_position3] = 'X'

# else:

# input("Position invalid. \nPlease select a new position for your third ship.")

# #Placer bateau 4

# ship\_position4 = input("Please enter a position to place your fourth ship: ")

# for i in board:

# if board[ship\_position]:

# board[ship\_position4] = 'X'

# else:

# input("Position invalid. \nPlease select a new position for your fourth ship.")

# #Placer bateau 5

# ship\_position5 = input("Please enter a position to place your fifth ship: ")

# for i in board:

# if board[ship\_position]:

# board[ship\_position5] = 'X'

# else:

# input("Position invalid. \nPlease select a new position for your fifth ship.")

print(board) #affichage du plateau

turn\_board\_list = list(board.values())

print(turn\_board\_list) #affichage de la liste

win\_condition = False #création d'une condition pour lancer la partie

while win\_condition == False: #lancement de la boucle de jeu

if "X" not in turn\_board\_list: #condition selon laquelle la partie s'arrête

print("\nIt’s a win! Congratz!") #plus aucun bateau = game over

break

else:

new\_turn\_shoot = (input("Pick a position to fire at: ")) #si la partie n'est pas finie : demande de coordonnées pour le tir

if new\_turn\_shoot in board: #accès à la clé correspondante aux coordonnées indiquées par le joeur

if board[new\_turn\_shoot] == "X": #si valeur associée à la clé = bateau

board.update({new\_turn\_shoot:"H"}) #transforme valeur en "H" pour indiquer "HIT"

print("\nIt's a HIT! :) ")

elif board[new\_turn\_shoot] == "O": #si valeur = case vide

board.update({new\_turn\_shoot:"M"}) #transforme value en "M" pour indiquer "miss"

print("\nIt's a MISS! :( ")

else:

print("\nWATCH OUT! You've already tried this position!!!") #si valeur n'est ni bateau, ni vide elle a forcément été essayée auparavant

turn\_board\_list = list(board.values()) #mise à jour du board après le tour

print("Your progress so far:\n", turn\_board\_list) #affichage du board pour le joueur

Dimanche 12h55:

v0.6 - Board display + error on shoot try if position unavailable

def display\_gameboard(any\_list):

start\_slice = 0

end\_slice = 10

for i in range(0,10):

print("[",i,"]" " ", any\_list[start\_slice:end\_slice])

start\_slice += 10

end\_slice += 10

def display\_gameboard\_rows(any\_list):

start\_slice = 0

end\_slice = 10

for i in range(0,1):

print("\n ", any\_list[start\_slice:end\_slice],"\n")

# start\_slice += 10

# end\_slice += 10

board\_rows = ["A", "B","C","D","E","F","G","H","I","J"]

board = {

"A1": "O","A2": "O","A3": "O","A4": "O","A5": "O","A6": "O","A7": "O","A8": "O","A9": "O","A10": "O",

"B1": "O","B2": "O","B3": "O","B4": "O","B5": "O","B6": "O","B7": "O","B8": "O","B9": "O","B10": "O",

"C1": "O","C2": "O","C3": "O","C4": "O","C5": "O","C6": "O","C7": "O","C8": "O","C9": "O","C10": "O",

"D1": "O","D2": "O","D3": "O","D4": "O","D5": "O","D6": "O","D7": "O","D8": "O","D9": "O","D10": "O",

"E1": "O","E2": "O","E3": "O","E4": "O","E5": "O","E6": "O","E7": "O","E8": "O","E9": "O","E10": "O",

"F1": "O","F2": "O","F3": "O","F4": "O","F5": "O","F6": "O","F7": "O","F8": "O","F9": "O","F10": "O",

"G1": "O","G2": "O","G3": "O","G4": "O","G5": "O","G6": "O","G7": "O","G8": "O","G9": "O","G10": "O",

"H1": "O","H2": "O","H3": "O","H4": "O","H5": "O","H6": "O","H7": "O","H8": "O","H9": "O","H10": "O",

"I1": "O","I2": "O","I3": "O","I4": "O","I5": "O","I6": "O","I7": "O","I8": "O","I9": "O","I10": "O",

"J1": "O","J2": "O","J3": "O","J4": "O","J5": "O","J6": "O","J7": "O","J8": "O","J9": "O","J10": "O",

}

#Placer bateau 1

ship\_position1 = input("Please select a position to place your first ship: ")

first\_boat\_placed = False #bateau 1 n'est pas placé

while first\_boat\_placed == False: #tant que bateau 1 n'est pas placé

if ship\_position1 in board: #si position indiquée est bien valide et présente sur le plateau

if board[ship\_position1] != 'X': #si la position n'est pas déjà occupée par un bateau

board[ship\_position1] = 'X' #position indiquée = bateau 1

first\_boat\_placed = True #bateau 1 placé => sortie de boucle de placement du bateau

else:

ship\_position1 = input("Position seems to be occupied by another boat.\nPlease select a new position for your first ship: ")

else:

ship\_position1 = input("The position you provided is not in the game board and therefore invalid.\nPlease select a new valid position for your first ship: ")

# #Placer bateau 2

ship\_position2 = input("Please select a position to place your second ship: ")

second\_boat\_placed = False #bateau 2 n'est pas placé

while second\_boat\_placed == False: #tant que bateau 2 n'est pas placé

if ship\_position2 in board: #si position indiquée est bien valide et présente sur le plateau

if board[ship\_position2] != 'X': #si la position n'est pas déjà occupée par un bateau

board[ship\_position2] = 'X' #position indiquée = bateau

second\_boat\_placed = True #bateau placé => sortie de boucle de placement du bateau

else:

ship\_position2 = input("Position seems to be occupied by another boat.\nPlease select a new position for your first ship: ")

else:

ship\_position2 = input("The position you provided is not in the game board and therefore invalid.\nPlease select a new valid position for your first ship: ")

# #Placer bateau 3

# ship\_position3 = input("Please enter a position to place your third ship: ")

# for i in board:

# if board[ship\_position]:

# board[ship\_position3] = 'X'

# else:

# input("Position invalid. \nPlease select a new position for your third ship.")

# #Placer bateau 4

# ship\_position4 = input("Please enter a position to place your fourth ship: ")

# for i in board:

# if board[ship\_position]:

# board[ship\_position4] = 'X'

# else:

# input("Position invalid. \nPlease select a new position for your fourth ship.")

# #Placer bateau 5

# ship\_position5 = input("Please enter a position to place your fifth ship: ")

# for i in board:

# if board[ship\_position]:

# board[ship\_position5] = 'X'

# else:

# input("Position invalid. \nPlease select a new position for your fifth ship.")

print(board) #affichage du plateau

turn\_board\_list = list(board.values())

print(turn\_board\_list) #affichage de la liste

win\_condition = False #création d'une condition pour lancer la partie

while win\_condition == False: #lancement de la boucle de jeu

if "X" not in turn\_board\_list: #condition selon laquelle la partie s'arrête

print("\nIt’s a win! Congratz!") #plus aucun bateau = game over

break

else:

new\_turn\_shoot = (input("\nPick a position to fire at: ")) #si la partie n'est pas finie : demande de coordonnées pour le tir

if new\_turn\_shoot in board: #accès à la clé correspondante aux coordonnées indiquées par le joeur

if board[new\_turn\_shoot] == "X": #si valeur associée à la clé = bateau

board.update({new\_turn\_shoot:"H"}) #transforme valeur en "H" pour indiquer "HIT"

print("\nIt's a HIT! :D ")

elif board[new\_turn\_shoot] == "O": #si valeur = case vide

board.update({new\_turn\_shoot:"M"}) #transforme value en "M" pour indiquer "miss"

print("\nIt's a MISS! T\_T")

else:

print("\nWATCH OUT! You've already tried this position!!!") #si valeur n'est ni bateau, ni vide elle a forcément été essayée auparavant

else:

input("The position you provided is not in the game board and therefore invalid.\nPlease select a new valid posiiton to shoot at: ")

# counter = 0

# for key, value in board.items():

# print(value, end=' ')

# counter += 1

# if counter == 10:

# print()

# counter = 0

# turn\_board = board

# print(board)

turn\_board\_list = list(board.values()) #mise à jour du board après le tour

display\_gameboard\_rows(board\_rows)

display\_gameboard(turn\_board\_list)

# print("Your progress so far:\n", turn\_board\_list) #affichage du board pour le joueur

Dimanche 15h40: plain\_board

def display\_gameboard(any\_list):

start\_slice = 0

end\_slice = 10

for i in range(1,11):

if i < 10:

print("[",i,"]" " ", any\_list[start\_slice:end\_slice])

else:

print("[",i,"]" " ", any\_list[start\_slice:end\_slice])

start\_slice += 10

end\_slice += 10

def display\_gameboard\_rows(any\_list):

start\_slice = 0

end\_slice = 10

for i in range(0,1):

print("\n ", any\_list[start\_slice:end\_slice],"\n")

board\_rows = ["1", "2","3","4","5","6","7","8","9","10"]

board = {

"A1": "O","A2": "O","A3": "O","A4": "O","A5": "O","A6": "O","A7": "O","A8": "O","A9": "O","A10": "O",

"B1": "O","B2": "O","B3": "O","B4": "O","B5": "O","B6": "O","B7": "O","B8": "O","B9": "O","B10": "O",

"C1": "O","C2": "O","C3": "O","C4": "O","C5": "O","C6": "O","C7": "O","C8": "O","C9": "O","C10": "O",

"D1": "O","D2": "O","D3": "O","D4": "O","D5": "O","D6": "O","D7": "O","D8": "O","D9": "O","D10": "O",

"E1": "O","E2": "O","E3": "O","E4": "O","E5": "O","E6": "O","E7": "O","E8": "O","E9": "O","E10": "O",

"F1": "O","F2": "O","F3": "O","F4": "O","F5": "O","F6": "O","F7": "O","F8": "O","F9": "O","F10": "O",

"G1": "O","G2": "O","G3": "O","G4": "O","G5": "O","G6": "O","G7": "O","G8": "O","G9": "O","G10": "O",

"H1": "O","H2": "O","H3": "O","H4": "O","H5": "O","H6": "O","H7": "O","H8": "O","H9": "O","H10": "O",

"I1": "O","I2": "O","I3": "O","I4": "O","I5": "O","I6": "O","I7": "O","I8": "O","I9": "O","I10": "O",

"J1": "O","J2": "O","J3": "O","J4": "O","J5": "O","J6": "O","J7": "O","J8": "O","J9": "O","J10": "O",

}

#Ajout Ismael 1 - Ajout d'un plateau vierge pour que le joueur ne voit pas les bateaux placés

plain\_board = {

"A1": "O","A2": "O","A3": "O","A4": "O","A5": "O","A6": "O","A7": "O","A8": "O","A9": "O","A10": "O",

"B1": "O","B2": "O","B3": "O","B4": "O","B5": "O","B6": "O","B7": "O","B8": "O","B9": "O","B10": "O",

"C1": "O","C2": "O","C3": "O","C4": "O","C5": "O","C6": "O","C7": "O","C8": "O","C9": "O","C10": "O",

"D1": "O","D2": "O","D3": "O","D4": "O","D5": "O","D6": "O","D7": "O","D8": "O","D9": "O","D10": "O",

"E1": "O","E2": "O","E3": "O","E4": "O","E5": "O","E6": "O","E7": "O","E8": "O","E9": "O","E10": "O",

"F1": "O","F2": "O","F3": "O","F4": "O","F5": "O","F6": "O","F7": "O","F8": "O","F9": "O","F10": "O",

"G1": "O","G2": "O","G3": "O","G4": "O","G5": "O","G6": "O","G7": "O","G8": "O","G9": "O","G10": "O",

"H1": "O","H2": "O","H3": "O","H4": "O","H5": "O","H6": "O","H7": "O","H8": "O","H9": "O","H10": "O",

"I1": "O","I2": "O","I3": "O","I4": "O","I5": "O","I6": "O","I7": "O","I8": "O","I9": "O","I10": "O",

"J1": "O","J2": "O","J3": "O","J4": "O","J5": "O","J6": "O","J7": "O","J8": "O","J9": "O","J10": "O",

}

#Placer bateau 1

ship\_position1 = input("Please select a position to place your first ship: ")

first\_boat\_placed = False #bateau 1 n'est pas placé

while first\_boat\_placed == False: #tant que bateau 1 n'est pas placé

if ship\_position1 in board: #si position indiquée est bien valide et présente sur le plateau

if board[ship\_position1] != 'X': #si la position n'est pas déjà occupée par un bateau

board[ship\_position1] = 'X' #position indiquée = bateau 1

first\_boat\_placed = True #bateau 1 placé => sortie de boucle de placement du bateau

else:

ship\_position1 = input("Position seems to be occupied by another boat.\nPlease select a new position for your first ship: ")

else:

ship\_position1 = input("The position you provided is not in the game board and therefore invalid.\nPlease select a new valid position for your first ship: ")

# #Placer bateau 2

ship\_position2 = input("Please select a position to place your second ship: ")

second\_boat\_placed = False #bateau 2 n'est pas placé

while second\_boat\_placed == False: #tant que bateau 2 n'est pas placé

if ship\_position2 in board: #si position indiquée est bien valide et présente sur le plateau

if board[ship\_position2] != 'X': #si la position n'est pas déjà occupée par un bateau

board[ship\_position2] = 'X' #position indiquée = bateau

second\_boat\_placed = True #bateau placé => sortie de boucle de placement du bateau

else:

ship\_position2 = input("Position seems to be occupied by another boat.\nPlease select a new position for your first ship: ")

else:

ship\_position2 = input("The position you provided is not in the game board and therefore invalid.\nPlease select a new valid position for your first ship: ")

# #Placer bateau 3

# ship\_position3 = input("Please enter a position to place your third ship: ")

# for i in board:

# if board[ship\_position]:

# board[ship\_position3] = 'X'

# else:

# input("Position invalid. \nPlease select a new position for your third ship.")

# #Placer bateau 4

# ship\_position4 = input("Please enter a position to place your fourth ship: ")

# for i in board:

# if board[ship\_position]:

# board[ship\_position4] = 'X'

# else:

# input("Position invalid. \nPlease select a new position for your fourth ship.")

# #Placer bateau 5

# ship\_position5 = input("Please enter a position to place your fifth ship: ")

# for i in board:

# if board[ship\_position]:

# board[ship\_position5] = 'X'

# else:

# input("Position invalid. \nPlease select a new position for your fifth ship.")

turn\_board\_list = list(board.values())

plain\_board\_list = list(plain\_board.values()) # Ajout ISMAËL 2

# print(board) #affichage du plateau avant le premier tour pour vérification

# print(turn\_board\_list) #affichage de la liste avant le premier tour pour vérification

win\_condition = False #création d'une condition pour lancer la partie

shot\_coordinates = False

while win\_condition == False: #lancement de la boucle de jeu

if "X" not in turn\_board\_list: #condition selon laquelle la partie s'arrête

print("\nYou've WON the game!\nCONGRATULATIONS!") #plus aucun bateau = game over

break

else:

new\_turn\_shot = (input("\nPick a position to fire at: ")) #si la partie n'est pas finie : demande de coordonnées pour le tir

new\_turn\_shot = new\_turn\_shot.capitalize()

if new\_turn\_shot in board: #accès à la clé correspondante aux coordonnées indiquées par le joeur

shot\_coordinates = True

if board[new\_turn\_shot] == "X": #si valeur associée à la clé = bateau

board.update({new\_turn\_shot:"H"}) #transforme valeur en "H" pour indiquer "HIT"

plain\_board.update({new\_turn\_shot:"H"}) #Ajout ISMAËL 5 transforme valeur en "H" pour indiquer "HIT" sur le tableau vierge

print("\nIt's a HIT! :D ")

elif board[new\_turn\_shot] == "O": #si valeur = case vide

board.update({new\_turn\_shot:"M"}) #transforme value en "M" pour indiquer "miss"

plain\_board.update({new\_turn\_shot:"M"}) # Ajout ISMAËL 6 #transforme value en "M" pour indiquer "miss"

print("\nIt's a MISS! T\_T")

else:

print("\nWATCH OUT! You've already tried this position !!!") #si valeur n'est ni bateau, ni vide elle a forcément été essayée auparavant

else:

print("The position you provided is not in the game board and therefore invalid.\nPlease select a new valid position to fire at.")

# turn\_board = board

# print(board)

turn\_board\_list = list(board.values()) #mise à jour du board après le tour

plain\_board\_list = list(plain\_board.values()) #Ajout ISMAËL 3 #mise à jour du board après le tour sur le tableau vierge

display\_gameboard\_rows(board\_rows)

#display\_gameboard(turn\_board\_list)

display\_gameboard(plain\_board\_list) #Ajout ISMAËL 4 # affichage du tableau vierge à partir de la fonction display\_gameboard

# print("Your progress so far:\n", turn\_board\_list) #affichage du board pour le joueur