CS 160 CS Orientation

More Lists in Python...



Odds and Ends

• Assignment #9???

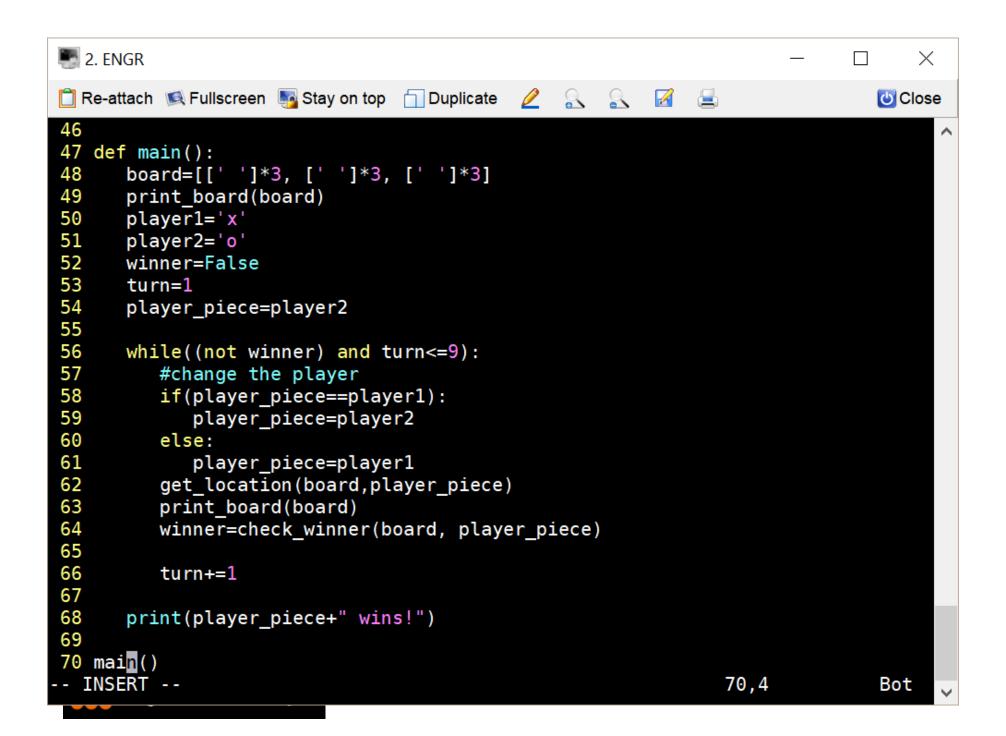


Finish Tic-tac-toe...



```
2. ENGR
                                                                                \times
📋 Re-attach 📧 Fullscreen 🏽 🥦 Stay on top 👘 Duplicate 🛛 🤌
                                              Close
                                                      1
 1 def print board(b):
 2
      print(b[0][0]+'|'+b[0][1]+'|'+b[0][2])
 3
      print('----')
      print(b[1][0]+'|'+b[1][1]+'|'+b[1][2])
 4
      print('----')
 5
      print(b[2][0]+'|'+b[2][1]+'|'+b[2][2])
 6
 7
 9 #
     Description: gets the location from the player and places the players
10 #
                  piece at that location
11 # Parameters: game board with spaces, 'x's, or 'o's, the player's piece
12 # Pre-conditions: The player's piece contains either a x or o character
13 #
     Post-condition: The board contains a space replaced with the player's
14 #
                      piece. (This means this function must check that row/col
15 #
                      contains a space before placing the piece on the board!)
16 # Returns: none
17 ######
18
19 def get location(b,p):
20
      #this should be in a loop while b[row][col] is not a space to satisfy
21
      #the postcondition
      row=int(input("Enter the row, 0-2: "))
22
      col=int(input("Enter the col, 0-2: "))
23
24
      b[row][col]=p
25
-- INSERT --
                                                              1,1
                                                                            Top
 UICYUII STALE UIIIVEISILY
```

```
E 2. ENGR
                                                                                   \times
                                                                             \square
📋 Re-attach 📧 Fullscreen 🌆 Stay on top 📋 Duplicate 🛛 🖉 🛛 🔬 🔗
                                                                               Close
25
                                                                                     ~
26 def check winner(b,p):
27
      #check horizontal
28
      for x in range(3):
29
         if(b[x][0]==p and b[x][1]==p and b[x][2]==p):
            return True
30
31
32
      #check vertical
33
      for x in range(3):
34
         if(b[0][x]==p and b[1][x]==p and b[2][x]==p):
35
            return True
36
37
      #check diagonal
38
      if(b[0][0]==p and b[1][1]==p and b[2][2]==p):
39
          return True
      if(b[0][2]==p and b[1][1]==p and b[2][0]==p):
40
         return True
41
42
43
      #return false if the horizontal, vertical, or diagonal directiosn do
44
      #not have 3 in a row.
45
      return False
46
47 def main():
      board=[[' ']*3, [' ']*3, [' ']*3]
48
      print board(board)
49
-- INSERT --
                                                                49,4
                                                                               53%
      UICYUII SLALE UIIIVEISILY
```



How do we change the travel program to use a 2-d array?

