Milestone Project: Walkthrough Steps Workbook

Below is a set of steps for you to follow to try to create the Tic Tac Toe Milestone Project game!

Some suggested tools before you get started:

To take input from a user:

```
player1 = input("Please pick a marker 'X' or 'O'")
```

Note that input() takes in a string. If you need an integer value, use

```
position = int(input('Please enter a number'))
```

To clear the screen between moves:

```
from IPython.display import clear_output
clear_output()
```

Note: Just copy paste the above mentioned "clear output" code for clearing the last output. It will help you get a Clean Output in the next step.
Note that clear_output() will only work in jupyter. To clear the screen in other IDEs, consider:

```
print('\n'*100)
```

This scrolls the previous board up out of view. Now on to the program!

Step 1: Write a function that can print out a board. Set up your board as a list, where each index 1-9 corresponds with a number on a number pad, so you get a 3 by 3 board representation.

In [1]:
```
from IPython.display import clear_output

def display_board(board):
    clear_output()
    print(board[7] + '|' + board[8] + '|' + board[9])
    print(board[4] + '|' + board[5] + '|' + board[6])
    print(board[1] + '|' + board[2] + '|' + board[3])
    pass
```

TEST Step 1: run your function on a test version of the board list, and make adjustments as necessary

In [2]:
```
test_board = ['#','X','O','X','O','X','O','X','O','X']
display_board(test_board)
```

```
X|O|X
O|X|O
X|O|X
```

Step 2: Write a function that can take in a player input and assign their marker as 'X' or 'O'. Think about using *while* loops to continually ask until you get a correct answer.

In [3]:
```
def player_input():
    marker = ''
    while marker != 'x' and marker !='o':
        marker = input('playa 1, choose x or o: ')
    playa1 = marker
    if playa1 == 'x':
        playa2 = 'o'
    else:
        playa2 = 'x'
    return (playa1, playa2)
    pass
```

TEST Step 2: run the function to make sure it returns the desired output

In [ ]:
```
player_input()
```

Step 3: Write a function that takes in the board list object, a marker ('X' or 'O'), and a desired position (number 1-9) and assigns it to the board.

In [ ]:
```
def place_marker(board, marker, position):
    board[position] = marker
    test_board
    pass
```

TEST Step 3: run the place marker function using test parameters and display the modified board

In [ ]:
```
place_marker(test_board,'$',8)
display_board(test_board)
```

Step 4: Write a function that takes in a board and a mark (X or O) and then checks to see if that mark has won.

In [ ]:
```
def win_check(board, mark):
    return ((board[4] == board[5] == board[6] == mark) or
    (board[7] == board[8] == board[9] == mark) or
    (board[1] == board[2] == board[3] == mark) or
    (board[1] == board[4] == board[7] == mark) or
    (board[2] == board[5] == board[8] == mark) or
    (board[3] == board[6] == board[9] == mark) or
    (board[7] == board[5] == board[3] == mark) or
    (board[9] == board[5] == board[1] == mark))
    pass
```

TEST Step 4: run the win_check function against our test_board - it should return True

In [ ]:
```
win_check(test_board,'X')
```

Step 5: Write a function that uses the random module to randomly decide which player goes first. You may want to lookup random.randint() Return a string of which player went first.

In [ ]:
```
import random

def choose_first():
    flip = random.randint(0,1)
    if flip == 0:
        return 'Player 1'
    else:
        return 'Player 2'
    pass
```

Step 6: Write a function that returns a boolean indicating whether a space on the board is freely available.

In [ ]:
```
def space_check(board, position):
    return board[position] == ' '
    pass
```

Step 7: Write a function that checks if the board is full and returns a boolean value. True if full, False otherwise.

In [ ]:
```
def full_board_check(board):
    for i in range(1,10):
        if space_check(board, i):
            return False
    return True #board is full#
    pass
```

Step 8: Write a function that asks for a player's next position (as a number 1-9) and then uses the function from step 6 to check if it's a free position. If it is, then return the position for later use.

In [ ]:
```
def player_choice(board):
    position = 0
    while position not in [1,2,3,4,5,6,7,8,9] or not space_check(board, positi
on):
        position = int(input('Choose a position 1-9 '))

    return position
    pass
```

Step 9: Write a function that asks the player if they want to play again and returns a boolean True if they do want to play again.

In [ ]:
```
def replay():
    print('Play again? enter Y or N ')
    return input().lower().startswith('y')

    pass
```

Step 10: Here comes the hard part! Use while loops and the functions you've made to run the game!

In [ ]:
```
print('Welcome to Tic Tac Toe!')

while True:
    # Set the game up here (board, who first, choose markers X,O)
    the_board = [' '] * 10
    playa1_marker, playa2_marker = player_input()

    turn = choose_first()
    print(turn + 'will go first')

    play_game = input('Ready to play? y or n')
    if play_game == 'y':
        game_on = True
    else:
        game_on = False


   # Play the game
    while game_on:
        if turn == 'Player 1': #Player 1 Turn

            display_board(the_board)#show the board
            position = player_choice(the_board)#choose position
            place_marker(the_board, playa1_marker, position)

            #check if they won
            if win_check(the_board, playa1_marker):
                display_board(the_board)
                print('Player 1 has won!!')
                game_on = False
            #check if its a tie
            else:
                if full_board_check(the_board):
                    display_board(the_board)
                    print('tie game')
                    break
                else:
                    turn = 'Player 2' #no tie and no win  next player turn
        else:
             #show the board  # Player2's turn.
            display_board(the_board)
            position = player_choice(the_board)
            place_marker(the_board, playa2_marker, position)

            #check if they won
            if win_check(the_board, playa2_marker):
                display_board(the_board)
                print('Player 2 has won!!')
                game_on = False
            #check if its a tie
            else:
                if full_board_check(the_board):
                    display_board(the_board)
                    print('tie game')
                    break
                else:
```

```
                    turn = 'Player 1' #no tie and no win  next player turn


#if not replay():
    if not replay():
        break
```

Good Job!