**Practice Of Project By LetsUpgrarde**

**from IPython.display import clear\_output**

**def display\_board(board):**

**clear\_output() # Remember, this only works in jupyter!**

**print(' | |')**

**print(' ' + board[7] + ' | ' + board[8] + ' | ' + board[9])**

**print(' | |')**

**print('-----------')**

**print(' | |')**

**print(' ' + board[4] + ' | ' + board[5] + ' | ' + board[6])**

**print(' | |')**

**print('-----------')**

**print(' | |')**

**print(' ' + board[1] + ' | ' + board[2] + ' | ' + board[3])**

**print(' | |')**

**test\_board = ['#','X','0','X','0', 'X', '0', 'X', '0','X']**

**display\_board(test\_board)**

**def player\_input():**

**marker = ''**

**while not (marker == 'X' or marker == '0'):**

**marker = input('Player 1: Do you want to be X or O? ').upper()**

**if marker == 'X':**

**return ('X', '0')**

**else:**

**return ('0', 'X')**

**Player 1: Do you want to be X or O? X**

**def place\_marker(board, marker, position):**

**board[position] = marker**

**place\_marker(test\_board, '$', 8)**

**display\_board(test\_board)**

**def win\_check(board, mark):**

**return ((board[7] == mark and board[8] == mark and board[9] == mark) or # accross the top(board[4] == mark an**

**d board[5] == mark and board[6] == mark) or # across the middle**

**(board[4] == mark and board[5] ==mark and board[6] == mark ) or**

**(board[1] == mark and board[2] == mark and board[3] == mark) or # across the bottom**

**(board[7] == mark and board[4] == mark and board[1] == mark) or # down the middle**

**(board[8] == mark and board[5] == mark and board[2] == mark) or # down the middle**

**(board[9] == mark and board[6] == mark and board[3] == mark) or # down the right side**

**(board[7] == mark and board[5] == mark and board[3] == mark) or # diagonal**

**(board[9] == mark and board[5] == mark and board[1] == mark)) # diagonal**

**win\_check(test\_board, 'X')**

**import random**

**def choose\_first():**

**if random.randint(0, 1) == 0:**

**return 'Player 2'**

**else:**

**return 'Player 1'**

**def space\_check(board, position):**

**return board[position] == ' '**

**def full\_board\_check(board):**

**for i in range(1,10):**

**if space\_check(board, i):**

**return False**

**return True**

**def player\_choise(board):**

**position = 0**

**while position not in [1,2,3,4,5,6,7,8,9] or not space\_check(board, position):**

**position = int(input('Choose your next position : (1-9) '))**

**return position**

**def reply():**

**return input('Do you want to play again ? Enter Yes or No: ').lower().startswith('y')**