FinishedWorkingVersion_2

February 26, 2018

1 Finished Working Version 3

1.1 How to Play

There are two players for the game.

The goal of the game is to get three noughts or crosses aligned within a 3x3 grid.

This can occur horizontally, vertically or diagonally.

If there no noughts or crosses aligned, the players have drawed and the game starts again.

```
In [3]: import random # For the AI to select placements within the board
        Player_1 = 'X'
        AI = 'O'

        def clear_board():
            board = [" "," "," "," "," "," "," "," "," "," "] # Creates empty spaces for the boa
            return board

        def print_board(board): # Prints the board in a grid format
            print('  {}  | {} |  {}  '.format(board[1],board[2],board[3]))
            print('---------------')
            print('  {}  | {} |  {}  '.format(board[4],board[5],board[6]))
            print('---------------')
            print('  {}  | {} |  {}  '.format(board[7],board[8],board[9]))

        def winner(char,board):
            if (board[1] == board[2] == board[3] == char or
              board[4] == board[5] == board[6] == char or
              board[7] == board[8] == board[9] == char or # Checks horizontal wins

              board[1] == board[4] == board[7] == char or
              board[2] == board[5] == board[8] == char or
              board[3] == board[6] == board[9] == char or # Check vertical wins

              board[1] == board[5] == board[9] == char or # Check diagonal wins
```

```
      board[3] == board[5] == board[9] == char):
        return True
    else:
        return False
    pass

def tie(Player_1,AI,board):
            if (board[1] != " " and board[2] !=" " and board[3] !=" " and board[4] !=" "
                return True
            else:
                print("It is a Draw!!!")
                return False
            pass

def player_win(Player_1,AI,board): # Check if one of the players have consecutive charac
    if winner(Player_1,board):                # Prints out which player wins
        print("Player 1 wins!!!")
        return
    elif winner(AI,board):
        print("AI 2 wins!!!")
        return
    else:
        return

def Game():

    global board
    board = clear_board() # Create a clear board
    print("---------- START -----------")
    while not player_win(Player_1,AI,board): # if there is no winner the game will conti
        print("Player One:")
        placement = int(input("Select a position"))
        board[placement] = Player_1
        print_board(board)
        player_win(Player_1,AI,board)
        print("Computer Turn:")
        placement = random.randint(1,10)
        board[placement] = AI
        print_board(board)
        player_win(Player_1,AI,board)


Game()
```

```
---------- START -----------
Player One:
Select a position5
     |   |
---------------
     | X |
---------------
     |   |
Computer Turn:
     |   |
---------------
     | X |
---------------
  O  |   |
Player One:
Select a position3
     |   |  X
---------------
     | X |
---------------
  O  |   |
Computer Turn:
     |   |  X
---------------
     | X |
---------------
  O  | O |
Player One:
Select a position9
     |   |  X
---------------
     | X |
---------------
  O  | O |  X
Player 1 wins!!!
Computer Turn:
     |   |  X
---------------
     | O |
---------------
  O  | O |  X
Player One:
Select a position6
     |   |  X
---------------
     | O |  X
---------------
  O  | O |  X
Player 1 wins!!!
Computer Turn:
    ---------------------------------------------------------------------------

    IndexError                                Traceback (most recent call last)

    <ipython-input-3-494618876555> in <module>()
     66
     67
---> 68 Game()


    <ipython-input-3-494618876555> in Game()
     61         print("Computer Turn:")
     62         placement = random.randint(1,10)
---> 63         board[placement] = AI
     64         print_board(board)
     65         player_win(Player_1,AI,board)


    IndexError: list assignment index out of range
```