Artificial Intelligence Lab

TIC TAC TOE Game

Name: Ahmed Kasteer

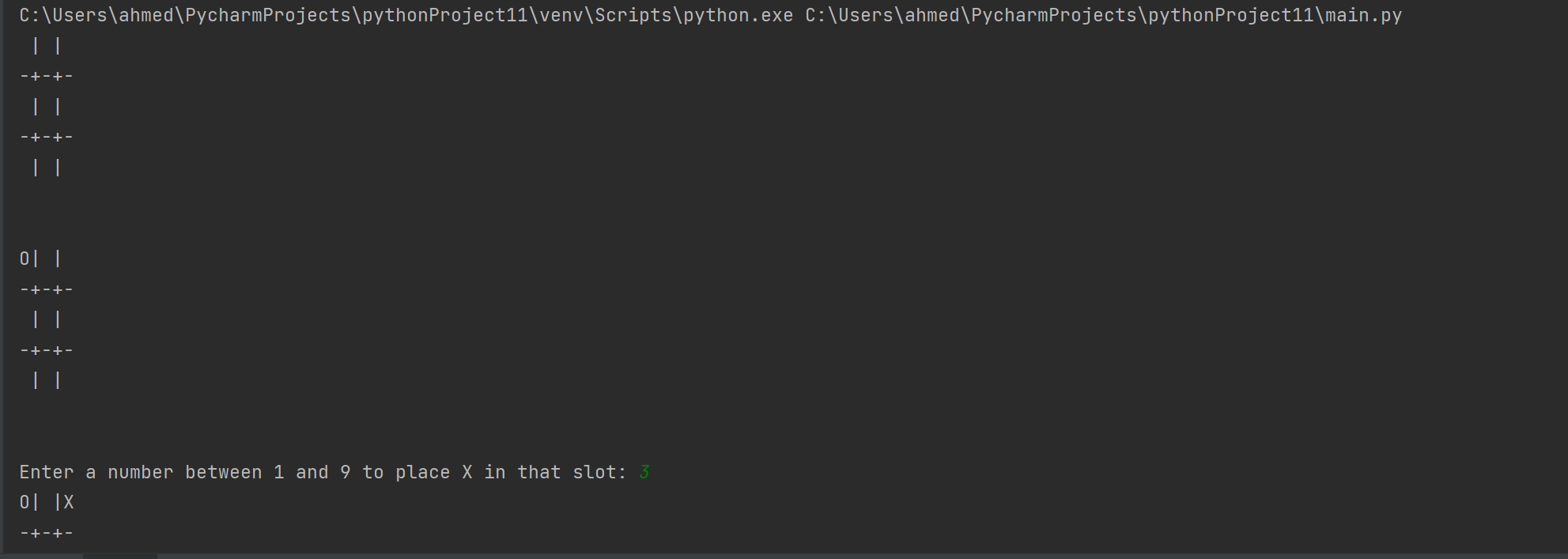
Roll Number: 20F-0336

Section: 6-C

Code:

# Define Board  
board = {  
 1: ' ', 2: ' ', 3: ' ',  
 4: ' ', 5: ' ', 6: ' ',  
 7: ' ', 8: ' ', 9: ' '}  
  
  
def printBoard(board):  
 print(board[1] + '|' + board[2] + '|' + board[3])  
 print('-+-+-')  
 print(board[4] + '|' + board[5] + '|' + board[6])  
 print('-+-+-')  
 print(board[7] + '|' + board[8] + '|' + board[9])  
 print('\n')  
  
  
def spaceVacant(position):  
 if board[position] == ' ':  
 return True  
 else:  
 return False  
  
  
def insertOnBoard(letter, position):  
 if spaceVacant(position):  
 board[position] = letter  
 printBoard(board)  
 if checkWithDraw():  
 print('Game Draw! ')  
 elif checkWin():  
 print('WON GAME! ')  
  
  
def checkWithDraw():  
 for key in board.keys():  
 if board[key] == ' ':  
 return False  
 return True  
  
  
def checkWin():  
 if board[1] == board[2] and board[1] == board[3] and board[1] != ' ':  
 return True  
 elif board[4] == board[5] and board[4] == board[6] and board[4] != ' ':  
 return True  
 elif board[7] == board[8] and board[7] == board[9] and board[7] != ' ':  
 return True  
 elif board[1] == board[5] and board[1] == board[9] and board[1] != ' ':  
 return True  
 elif board[3] == board[5] and board[3] == board[7] and board[3] != ' ':  
 return True  
 elif board[1] == board[4] and board[1] == board[7] and board[1] != ' ':  
 return True  
 elif board[2] == board[5] and board[2] == board[8] and board[2] != ' ':  
 return True  
 elif board[3] == board[6] and board[3] == board[9] and board[3] != ' ':  
 return True  
 else:  
 return False  
  
  
def userMove():  
 position = int(input("Enter a number between 1 and 9 to place X in that slot: "))  
 if position < 1 or position > 9:  
 print("Invalid input. Please enter a number between 1 and 9.")  
 userMove()  
 elif not spaceVacant(position):  
 print("Slot is already occupied. Please select an empty slot.")  
 userMove()  
 else:  
 insertOnBoard('X', position)  
  
  
def computerMove():  
 bestScore = float('-inf')  
 bestMove = None  
 for move in board.keys():  
 if board[move] == ' ':  
 board[move] = 'O'  
 score = minmax(board, False)  
 board[move] = ' '  
 if score > bestScore:  
 bestScore = score  
 bestMove = move  
 insertOnBoard('O', bestMove)  
  
  
def minmax(board, isMax):  
 if checkWin() and isMax:  
 return -1  
 elif checkWin() and not isMax:  
 return 1  
 elif checkWithDraw():  
 return 0  
  
 if isMax:  
 bestScore = float('-inf')  
 for move in board.keys():  
 if board[move] == ' ':  
 board[move] = 'O'  
 score = minmax(board, False)  
 board[move] = ' '  
 bestScore = max(score, bestScore)  
 return bestScore  
 else:  
 bestScore = float('inf')  
 for move in board.keys():  
 if board[move] == ' ':  
 board[move] = 'X'  
 score = minmax(board, True)  
 board[move] = ' '  
 bestScore = min(score, bestScore)  
 return bestScore  
  
  
# Press the green button in the gutter to run the script.  
if \_\_name\_\_ == '\_\_main\_\_':  
 printBoard(board)  
 computerMove()  
 while not checkWithDraw() and not checkWin():  
 userMove()  
 if not checkWithDraw() and not checkWin():  
 print("Computer move: ")  
 computerMove()  
 else:  
 break

Output:



Text

Description automatically generated