  1. # Tic Tac Toe

  2.

  3. import random

  4.

  5. def drawBoard(board):

  6.     # This function prints out the board that it was passed.

  7.

  8.     # "board" is a list of 10 strings representing the board (ignore index 0)

  9.     print('   |   |')

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

 11.     print('   |   |')

 12.     print('-----------')

 13.     print('   |   |')

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

 15.     print('   |   |')

 16.     print('-----------')

 17.     print('   |   |')

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

 19.     print('   |   |')

 20.

 21. def inputPlayerLetter():

 22.     # Lets the player type which letter they want to be.

 23.     # Returns a list with the player’s letter as the first item, and the computer's letter as the second.

 24.     letter = ''

 25.     while not (letter == 'X' or letter == 'O'):

 26.         print('Do you want to be X or O?')

 27.         letter = input().upper()

 28.

 29.     # the first element in the list is the player’s letter, the second is the computer's letter.

 30.     if letter == 'X':

 31.         return ['X', 'O']

 32.     else:

 33.         return ['O', 'X']

 34.

 35. def whoGoesFirst():

 36.     # Randomly choose the player who goes first.

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

 38.         return 'computer'

 39.     else:

 40.         return 'player'

 41.

 42. def playAgain():

 43.     # This function returns True if the player wants to play again, otherwise it returns False.

 44.     print('Do you want to play again? (yes or no)')

 45.     return input().lower().startswith('y')

 46.

 47. def makeMove(board, letter, move):

 48.     board[move] = letter

 49.

 50. def isWinner(bo, le):

 51.     # Given a board and a player’s letter, this function returns True if that player has won.

 52.     # We use bo instead of board and le instead of letter so we don’t have to type as much.

 53.     return ((bo[7] == le and bo[8] == le and bo[9] == le) or # across the top

 54.     (bo[4] == le and bo[5] == le and bo[6] == le) or # across the middle

 55.     (bo[1] == le and bo[2] == le and bo[3] == le) or # across the bottom

 56.     (bo[7] == le and bo[4] == le and bo[1] == le) or # down the left side

 57.     (bo[8] == le and bo[5] == le and bo[2] == le) or # down the middle

 58.     (bo[9] == le and bo[6] == le and bo[3] == le) or # down the right side

 59.     (bo[7] == le and bo[5] == le and bo[3] == le) or # diagonal

 60.     (bo[9] == le and bo[5] == le and bo[1] == le)) # diagonal

 61.

 62. def getBoardCopy(board):

 63.     # Make a duplicate of the board list and return it the duplicate.

 64.     dupeBoard = []

 65.

 66.     for i in board:

 67.         dupeBoard.append(i)

 68.

 69.     return dupeBoard

 70.

 71. def isSpaceFree(board, move):

 72.     # Return true if the passed move is free on the passed board.

 73.     return board[move] == ' '

 74.

 75. def getPlayerMove(board):

 76.     # Let the player type in their move.

 77.     move = ' '

 78.     while move not in '1 2 3 4 5 6 7 8 9'.split() or not isSpaceFree(board, int(move)):

 79.         print('What is your next move? (1-9)')

 80.         move = input()

 81.     return int(move)

 82.

 83. def chooseRandomMoveFromList(board, movesList):

 84.     # Returns a valid move from the passed list on the passed board.

 85.     # Returns None if there is no valid move.

 86.     possibleMoves = []

 87.     for i in movesList:

 88.         if isSpaceFree(board, i):

 89.             possibleMoves.append(i)

 90.

 91.     if len(possibleMoves) != 0:

 92.         return random.choice(possibleMoves)

 93.     else:

 94.         return None

 95.

 96. def getComputerMove(board, computerLetter):

 97.     # Given a board and the computer's letter, determine where to move and return that move.

 98.     if computerLetter == 'X':

 99.         playerLetter = 'O'

100.     else:

101.         playerLetter = 'X'

102.

103.     # Here is our algorithm for our Tic Tac Toe AI:

104.     # First, check if we can win in the next move

105.     for i in range(1, 10):

106.         copy = getBoardCopy(board)

107.         if isSpaceFree(copy, i):

108.             makeMove(copy, computerLetter, i)

109.             if isWinner(copy, computerLetter):

110.                 return i

111.

112.     # Check if the player could win on their next move, and block them.

113.     for i in range(1, 10):

114.         copy = getBoardCopy(board)

115.         if isSpaceFree(copy, i):

116.             makeMove(copy, playerLetter, i)

117.             if isWinner(copy, playerLetter):

118.                 return i

119.

120.     # Try to take one of the corners, if they are free.

121.     move = chooseRandomMoveFromList(board, [1, 3, 7, 9])

122.     if move != None:

123.         return move

124.

125.     # Try to take the center, if it is free.

126.     if isSpaceFree(board, 5):

127.         return 5

128.

129.     # Move on one of the sides.

130.     return chooseRandomMoveFromList(board, [2, 4, 6, 8])

131.

132. def isBoardFull(board):

133.     # Return True if every space on the board has been taken. Otherwise return False.

134.     for i in range(1, 10):

135.         if isSpaceFree(board, i):

136.             return False

137.     return True

138.

139.

140. print('Welcome to Tic Tac Toe!')

141.

142. while True:

143.     # Reset the board

144.     theBoard = [' '] \* 10

145.     playerLetter, computerLetter = inputPlayerLetter()

146.     turn = whoGoesFirst()

147.     print('The ' + turn + ' will go first.')

148.     gameIsPlaying = True

149.

150.     while gameIsPlaying:

151.         if turn == 'player':

152.             # Player’s turn.

153.             drawBoard(theBoard)

154.             move = getPlayerMove(theBoard)

155.             makeMove(theBoard, playerLetter, move)

156.

157.             if isWinner(theBoard, playerLetter):

158.                 drawBoard(theBoard)

159.                 print('Hooray! You have won the game!')

160.                 gameIsPlaying = False

161.             else:

162.                 if isBoardFull(theBoard):

163.                     drawBoard(theBoard)

164.                     print('The game is a tie!')

165.                     break

166.                 else:

167.                     turn = 'computer'

168.

169.         else:

170.             # Computer’s turn.

171.             move = getComputerMove(theBoard, computerLetter)

172.             makeMove(theBoard, computerLetter, move)

173.

174.             if isWinner(theBoard, computerLetter):

175.                 drawBoard(theBoard)

176.                 print('The computer has beaten you! You lose.')

177.                 gameIsPlaying = False

178.             else:

179.                 if isBoardFull(theBoard):

180.                     drawBoard(theBoard)

181.                     print('The game is a tie!')

182.                     break

183.                 else:

184.                     turn = 'player'

185.

186.     if not playAgain():

187.         break