

4/10/24

LAB-1

2) Implement a tic tac toe game using python

Pseudocode :-

function minimax (node, depth, isMaximizingPlayer):

if node is a terminal state:

return evaluate (node)

if isMaximizingPlayer:

bestValue = -infinity

for each child in node:

value = minimax (child, depth + 1, false)

bestValue = max (bestValue, value)

return bestValue

else:

bestValue = +infinity

for each child in node:

value = minimax (child, depth + 1, true)

bestValue = min (bestValue, value)

return bestValue



```

1 board = {1: ' ', 2: ' ', 3: ' ',
2           4: ' ', 5: ' ', 6: ' ',
3           7: ' ', 8: ' ', 9: ' '}
4
5 def printBoard(board):
6     print(board[1]+'|'+board[2]+'|'+board[3])
7     print('-+-+-')
8     print(board[4]+'|'+board[5]+'|'+board[6])
9     print('-+-+-')
10    print(board[7]+'|'+board[8]+'|'+board[9])
11    print('\n')
12
13 def spaceFree(pos):
14     if board[pos] == ' ':
15         return True
16     else:
17         return False
18
19 def checkWin():
20     if (board[1] == board[2] and board[1] == board[3] and board[1] != ' '):
21         return True
22     elif (board[4] == board[5] and board[4] == board[6] and board[4] != ' '):
23         return True
24     elif (board[7] == board[8] and board[7] == board[9] and board[7] != ' '):
25         return True
26     elif (board[1] == board[5] and board[1] == board[9] and board[1] != ' '):
27         return True
28     elif (board[3] == board[5] and board[3] == board[7] and board[3] != ' '):
29         return True
30     elif (board[1] == board[4] and board[1] == board[7] and board[1] != ' '):
31         return True
32     elif (board[2] == board[5] and board[2] == board[8] and board[2] != ' '):
33         return True
34     elif (board[3] == board[6] and board[3] == board[9] and board[3] != ' '):
35         return True

```

```
36     else:
37         return False
38
39 def checkMoveForWin(move):
40     if (board[1] == board[2] and board[1] == board[3] and board[1] == move):
41         return True
42     elif (board[4] == board[5] and board[4] == board[6] and board[4] == move):
43         return True
44     elif (board[7] == board[8] and board[7] == board[9] and board[7] == move):
45         return True
46     elif (board[1] == board[5] and board[1] == board[9] and board[1] == move):
47         return True
48     elif (board[3] == board[5] and board[3] == board[7] and board[3] == move):
49         return True
50     elif (board[1] == board[4] and board[1] == board[7] and board[1] == move):
51         return True
52     elif (board[2] == board[5] and board[2] == board[8] and board[2] == move):
53         return True
54     elif (board[3] == board[6] and board[3] == board[9] and board[3] == move):
55         return True
56     else:
57         return False
58
59 def checkDraw():
60     for key in board.keys():
61         if board[key] == ' ':
62             return False
63     return True
64
```

```
71     elif checkWin():
72         if letter == 'X':
73             print('Bot wins!')
74         else:
75             print('You win!')
76         return
77     else:
78         print('Position taken, please pick a different position.')
79         position = int(input('Enter new position: '))
80         insertLetter(letter, position)
81         return
82
83 player = 'O'
84 bot = 'X'
85
86 def playerMove():
87     position = int(input('Enter position for O: '))
88     insertLetter(player, position)
89     return
90
91 def compMove():
92     bestScore = -1000
93     bestMove = 0
94     for key in board.keys():
95         if board[key] == ' ':
96             board[key] = bot
97             score = minimax(board, False)
```



```
105 def minimax(board, isMaximizing):
106     if checkMoveForWin(bot):
107         return 1
108     elif checkMoveForWin(player):
109         return -1
110     elif checkDraw():
111         return 0
112
113     if isMaximizing:
114         bestScore = -1000
115         for key in board.keys():
116             if board[key] == ' ':
117                 board[key] = bot
118                 score = minimax(board, False)
119                 board[key] = ' '
120                 if score > bestScore:
121                     bestScore = score
122         return bestScore
123     else:
124         bestScore = 1000
125         for key in board.keys():
126             if board[key] == ' ':
127                 board[key] = player
128                 score = minimax(board, True)
129                 board[key] = ' '
130                 if score < bestScore:
131                     bestScore = score
132         return bestScore
133 print("Tanish M v")
134 print("1BM22CS302")
135 while not checkWin():
136     compMove()
137     playerMove()
```

Tanish M v

1BM22CS302

X| |

-+-+--

| |

-+-+--

| |

Enter position for 0: 2

X|O|

-+-+--

| |

-+-+--

| |

X|O|

-+-+--

X| |

-+-+--

| |

Enter position for 0: 7

X|O|

-+-+--

X| |

-+-+--

O| |

```
X|O|
-+-+
X|X|
-+-+
O| |
```

Enter position for O: 3

```
X|O|O
-+-+
X|X|
-+-+
O| |
```

```
X|O|O
-+-+
X|X|X
-+-+
O| |
```

Bot wins!