

```

1 # Name : Vasu Kalariya
2 # Roll : PE29
3 # AI lab Assi 2 (MinMax)
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
14 def spaceIsFree(position):                # checking for the space is free or not
15     if board[position] == ' ':
16         return True
17     else:
18         return False
19
20
21 def insertSymbol(letter, position):        # insert symbol at given number
22     if spaceIsFree(position):             # check for free space
23         board[position] = letter
24         printBoard(board)
25         if (checkDraw()):                 # check for Draw
26             print("Draw!")
27             exit()
28         if checkForWin():                 # check for WIN
29             if letter == 'X':
30                 print("AI wins!")
31                 exit()
32             else:
33                 print("Player wins!")
34                 exit()
35
36         return
37
38
39     else:                                  # space is already filled
40         print("Can't insert there!")
41         position = int(input("Please enter new position: "))
42         insertSymbol(letter, position)
43         return
44
45
46 def checkForWin():
47     if (board[1] == board[2] and board[1] == board[3] and board[1] != ' '):
48         return True
49     elif (board[4] == board[5] and board[4] == board[6] and board[4] != ' '):
50         return True
51     elif (board[7] == board[8] and board[7] == board[9] and board[7] != ' '):
52         return True
53     elif (board[1] == board[4] and board[1] == board[7] and board[1] != ' '):
54         return True
55     elif (board[2] == board[5] and board[2] == board[8] and board[2] != ' '):
56         return True
57     elif (board[3] == board[6] and board[3] == board[9] and board[3] != ' '):
58         return True
59     elif (board[1] == board[5] and board[1] == board[9] and board[1] != ' '):
60         return True

```

```
61     elif (board[7] == board[5] and board[7] == board[3] and board[7] != ' '):
62         return True
63     else:
64         return False
65
66
67 def checkWhichSymbolWon(symbol):
68     if board[1] == board[2] and board[1] == board[3] and board[1] == symbol:
69         return True
70     elif (board[4] == board[5] and board[4] == board[6] and board[4] == symbol):
71         return True
72     elif (board[7] == board[8] and board[7] == board[9] and board[7] == symbol):
73         return True
74     elif (board[1] == board[4] and board[1] == board[7] and board[1] == symbol):
75         return True
76     elif (board[2] == board[5] and board[2] == board[8] and board[2] == symbol):
77         return True
78     elif (board[3] == board[6] and board[3] == board[9] and board[3] == symbol):
79         return True
80     elif (board[1] == board[5] and board[1] == board[9] and board[1] == symbol):
81         return True
82     elif (board[7] == board[5] and board[7] == board[3] and board[7] == symbol):
83         return True
84     else:
85         return False
86
87
88 def checkDraw():
89     for key in board.keys():
90         if (board[key] == ' '):
91             return False
92     return True
93
94
95 def playerTurn():
96     position = int(input("Enter the position for 'O': "))
97     insertSymbol(player, position)
98     return
99
100
101 def compTurn():
102     bestScore = -800
103     bestMove = 0
104     for key in board.keys():
105         if (board[key] == ' '):
106             board[key] = AI
107             score = minimax(board, False)
108             board[key] = ' '
109             if (score > bestScore):
110                 bestScore = score
111                 bestMove = key
112
113     insertSymbol(AI, bestMove)
114     return
115
116
117 def minimax(board, isMaximizing):
118     if (checkWhichSymbolWon(AI)):
119         return 1
120     elif (checkWhichSymbolWon(player)):
```

```
121         return -1
122     elif (checkDraw()):
123         return 0
124
125     if (isMaximizing):                                     # trying to Maximize score
126         bestScore = -1000
127         for key in board.keys():
128             if (board[key] == ' '):
129                 board[key] = AI
130                 score = minimax(board, False)
131                 board[key] = ' '
132                 if (score > bestScore):
133                     bestScore = score
134         return bestScore
135
136     else:                                                  # trying to Minimize score at
137         next depth
138         bestScore = 1000
139         for key in board.keys():
140             if (board[key] == ' '):
141                 board[key] = player
142                 score = minimax(board, True)
143                 board[key] = ' '
144                 if (score < bestScore):
145                     bestScore = score
146         return bestScore
147
148 board = {1: ' ', 2: ' ', 3: ' ',
149          4: ' ', 5: ' ', 6: ' ',
150          7: ' ', 8: ' ', 9: ' '}
151
152
153
154 print("Positions are as follow:")
155 print("")
156 print("1, 2, 3 ")
157 print("4, 5, 6 ")
158 print("7, 8, 9 ")
159 print("\n")
160 player = 'O'
161 AI = 'X'
162 printBoard(board)
163
164 while not checkForWin():
165     playerTurn()
166     compTurn()
```