

```

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')
print("Vignesh B 1BM22CS326 ")
def spaceFree(pos):
    return board[pos] == ' '

def checkWin():
    win_conditions = [
        (1, 2, 3), (4, 5, 6), (7, 8, 9),
        (1, 4, 7), (2, 5, 8), (3, 6, 9),
        (1, 5, 9), (3, 5, 7)
    ]
    for condition in win_conditions:
        if board[condition[0]] == board[condition[1]] == board[condition[2]] != ' ':
            return True
    return False

def checkDraw():
    return all(board[key] != ' ' for key in board.keys())

def insertLetter(letter, position):
    if spaceFree(position):
        board[position] = letter
        printBoard(board)

        if checkDraw():
            print('The game is a draw!')
        elif checkWin():
            print(f'{letter} wins!')
        return

    else:
        print('Position taken, please pick a different position.')
        position = int(input('Enter new position: '))
        insertLetter(letter, position)

player = 'O'
bot = 'X'

```

```
def playerMove():
    while True:
        try:
            position = int(input('Enter position for 0 (1-9): '))
            if position in range(1, 10) and spaceFree(position):
                insertLetter(player, position)
                break
            else:
                print("Invalid position or position taken, please choose again.")
        except ValueError:
            print("Please enter a valid number between 1 and 9.")

def compMove():
    bestScore = -1000
    bestMove = 0
    for key in board.keys():
        if spaceFree(key):
            board[key] = bot
            score = minimax(board, False)
            board[key] = ' '
            if score > bestScore:
                bestScore = score
                bestMove = key

    insertLetter(bot, bestMove)

def minimax(board, isMaximizing):
    if checkWin():
        return 1 if isMaximizing else -1
    if checkDraw():
        return 0

    if isMaximizing:
        bestScore = -1000
        for key in board.keys():
            if spaceFree(key):
                board[key] = bot
                score = minimax(board, False)
                board[key] = ' '
                bestScore = max(bestScore, score)
        return bestScore
    else:
        bestScore = 1000
        for key in board.keys():
            if spaceFree(key):
                board[key] = player
                score = minimax(board, True)
                board[key] = ' '

```

```

        bestScore = min(bestScore, score)
    return bestScore

while True:
    compMove()
    if checkWin() or checkDraw():
        break
    playerMove()
    if checkWin() or checkDraw():
        break
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 spaceFree(pos):
    return board[pos] == ' '

def checkWin():
    win_conditions = [
        (1, 2, 3), (4, 5, 6), (7, 8, 9),
        (1, 4, 7), (2, 5, 8), (3, 6, 9),
        (1, 5, 9), (3, 5, 7)
    ]
    for condition in win_conditions:
        if board[condition[0]] == board[condition[1]] == board[condition[2]] != ' ':
            return True
    return False

def checkDraw():
    return all(board[key] != ' ' for key in board.keys())

def insertLetter(letter, position):
    if spaceFree(position):
        board[position] = letter
        printBoard(board)

        if checkDraw():
            print('The game is a draw!')
        elif checkWin():
            print(f'{letter} wins!')
    return

```

```

    return

    else:
        print('Position taken, please pick a different position.')
        position = int(input('Enter new position: '))
        insertLetter(letter, position)

player = 'O'
bot = 'X'

def playerMove():
    while True:
        try:
            position = int(input('Enter position for O (1-9): '))
            if position in range(1, 10) and spaceFree(position):
                insertLetter(player, position)
                break
            else:
                print("Invalid position or position taken, please choose again.")
        except ValueError:
            print("Please enter a valid number between 1 and 9.")

def compMove():
    bestScore = -1000
    bestMove = 0
    for key in board.keys():
        if spaceFree(key):
            board[key] = bot
            score = minimax(board, False)
            board[key] = ' '
            if score > bestScore:
                bestScore = score
                bestMove = key

    insertLetter(bot, bestMove)

def minimax(board, isMaximizing):
    if checkWin():
        return 1 if isMaximizing else -1
    if checkDraw():
        return 0

    if isMaximizing:
        bestScore = -1000
        for key in board.keys():
            if spaceFree(key):
                board[key] = bot
                score = minimax(board, False)
                board[key] = ' '

```

```

        bestScore = max(bestScore, score)
    return bestScore
else:
    bestScore = 1000
    for key in board.keys():
        if spaceFree(key):
            board[key] = player
            score = minimax(board, True)
            board[key] = ' '
            bestScore = min(bestScore, score)
    return bestScore

while True:
    compMove()
    if checkWin() or checkDraw():
        break
    playerMove()
    if checkWin() or checkDraw():
        break

```

... Vignesh B 1BM22CS326

```

| |
-+-+
|X|
-+-+
| |

```

Enter position for 0 (1-9): 1

```

0| |
-+-+
|X|
-+-+
| |

```

```

0| |
-+-+
|X|
-+-+
| |X

```

Enter position for 0 (1-9): 3

```

0| |0
-+-+
|X|
-+-+
| |X

```

```
0| |0
--+--
 |X|X
--+--
 | |X
```

Enter position for 0 (1-9): 2

```
0|0|0
--+--
 |X|X
--+--
 | |X
```

O wins!

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

Enter position for 0 (1-9):

Start coding or [generate](#) with AI.