

# Introduction to Artificial Intelligence

## Assignment 2

---

### MiniMax search for the 4X4 Tic-Tac-Toe game

The second assignment performs search on a game tree, where there are adversary agents each pursuing its own goal. The game you are requested to solve by the MiniMax algorithm is the 4X4 Tic-Tac-Toe game. The board game has 16 squares, there are 2 players playing in turns, and each player in its turn selects a square that is still empty and inserts its symbol in it. There are two categories of outcomes. One has a complete row, column, or diagonal taken by a single player. In this type of outcome, the player that has taken the 4-in-a-row wins and gets a point and the other player gets -1. The only other type of outcome has all squares taken, but no 4-in-a row. In this case the outcome is a draw, and each player gets 0.

Your program is required to perform MiniMax search in response to each step taken by the user, who is its opponent. The user performs the initial step, so let's call her Max. The program responds with its step, that is selected by the search program. The board is represented as a 4x4 matrix, and the user enters two integers (in range 0-3) to select its move. The program first marks the board accordingly, and then prints out its response (selected by running the search) by printing the resulting matrix of the currently selected squares. Such as follows:

X	X	○	
		○	

### **Note: starting randomly**

In order to have a lower-depth tree for the program search, the first step of the two players will be randomly selected. So, the first selection of the user is actually at depth 2. In order to enable comparison of results, the program can also accept this first step of the two players as input.

### **Note: MiniMax search**

The search algorithm is a recursive function that is composed of the Min function and the Max function, both returning a value and a selected move. The tail recursion returns at a leaf node. Please pay special attention to the fact that leaf nodes can be either a win/loss state or a tie. These two types of terminal nodes occur usually at different depths in the tree.

### **Output**

The output of your program must follow the following format, starting with the first random move of both players as in the example screen below

```
>
board:
X _ _ 0
_ _ _ _
_ _ _ _
_ _ _ _

choose row (0-3)
1
choose column (0-3)
1
board:
X _ _ 0
_ X _ _
_ _ _ _
_ _ _ _

AI plays...
board:
X _ _ 0
_ X _ _
_ _ _ _
_ _ _ 0

choose row (0-3)
4
choose column (0-3)
0
board:
X _ _ 0
_ X _ _
_ _ _ _
X _ _ 0

AI plays...
```

board:

```
X _ _ O
_ X _ O
_ _ _ _
X _ _ O
```

.....

DRAW/WIN/LOSE

number of visited nodes: 56494

>

## Deliverables

The program and code sent to the grader. A document describing your implementation. The document should also describe a couple of examples, demonstrating the run of, say, 2 steps of the play.

**Due date:** Tuesday, October 10, 2023

Enjoy !!!

