CE213 Artificial Intelligence Assignment (2021-2022)

DEADLINE: 11:59 am (noon), Tuesday 8th March 2022 (WEEK 23)

THE PROBLEM AND TASKS:

For this assignment you are asked to modify and extend a provided Java program using minimax search for playing Tic-Tac-Toe game on a 3x3 grid to a program for playing Tic-Tac-Toe game on a 5x5 grid. The wining criterion for playing Tic-Tac-Toe on a 5x5 grid is '5 in a line (row, column, or diagonal line)'. Tic-Tac-Toe is also called Noughts and Crosses. For details about the game, visit https://en.wikipedia.org/wiki/Tic-tac-toe.

The provided code is the one for Lab Exercise 2 at Your Own Time, which is available at https://moodle.essex.ac.uk/course/view.php?id=3651&section=16. Before doing this assignment, you are supposed to have run and understood the provided code, which implements minimax search that always returns values from endgame states. Your modified code needs to implement minimax search that can return values of game states at certain depth rather than endgame states by using a heuristic function for evaluating non-endgame states, because returning values from endgame states on a 5x5 grid would be unrealistic - too slow for the minimax search to return values for making a move.

Specifically, your main tasks are as follows:
1) To extend the Board class for playing Tic-Tac-Toe on a 5x5 grid, including its relevant methods;
2) To add a method for calculating the heuristic value of any given game state (board), which could be added in the Board class or the AI_Player class;
3) To modify the minimax method for returning values of non-endgame states rather than endgame states if the depth is too large (larger than a preset maximum depth, say, 6).

It is optional to use alpha-beta pruning in this assignment, which will not affect the marks for correctness but you may get a few extra marks for quality of your program in terms of its speed of making good moves.

You need to comment the modified and extended code. The general guidelines for code commenting are as follows: There are two types of comments: documentation comments and implementation comments. Documentation comments describe the semantics of a class or method. Good documentation comments should allow someone to use the class and its methods without having to read any source code. In contrast, implementation comments are used to clarify how a particular piece of code operates. You write implementation comments when you feel they are necessary (e.g., for this assignment, it is necessary to comment on the calculation of heuristic value of a game state (board) and important steps in the minimax search method).

In one of the Friday classes, we will discuss this assignment, especially how to evaluate non-endgame states with heuristic functions.

On rare occasions, if you have not learnt Java programming, the task of your assignment can be changed to create a Python program for playing Tic-Tac-Toe game on a 3x3 grid, which should implement minimax search and function in a similar way as the provided Java code. For this alternative, you should contact me to get my approval before Week 19.

SUBMISSION:

The assignment should be submitted through the online coursework submission system (FASER).

You should submit a single zip file with your registration number as the file name, containing all the source code files that comprise your modified and extended program for playing Tic-Tac-Toe on a 5x5 grid, with sufficient comments inside the code.

If you have been allowed to do this assignment in Python, then you should submit a single zip file with your registration number as the file name, containing all the source code files that comprise your program for playing Tic-Tac-Toe on a 3x3 grid, with sufficient comments inside the code.

ASSESSMENT CRITERIA:

Criteria	Mark	Comments
Correctness of the program (60%)	/60	
Quality of the program (its structure and efficiency/speed, etc.) (25%)	/25	
Clarity of the program and its commenting (15%)	/15	
Total	/100	

N.B. For each criterion, the following three parts of your work will equally make contributions to the marks you can get: 1) Extending the Board class for playing Tic-Tac-Toe on a 5x5 grid, including its relevant methods; 2) Adding a method for calculating the heuristic value of any given game state (board), which could be added in the Board class or the AI_Player class; 3) Modifying the minimax method for returning values of non-endgame states rather than endgame states if the depth is too large (larger than a preset maximum depth).

LATE SUBMISSION AND PLAGIARISM

Please refer to the Undergraduate Students' Handbook for details of the School policy regarding late submission and University regulations regarding academic offences:

This is an individual assignment. Please acknowledge clearly by commenting your program if any part of your program involves any contribution or help from your classmates or from Internet resources. Otherwise, use of other people's work in your assignment without proper acknowledgement and reference will be reported as an SAO case.

As this assignment is about modifying and extending the provided code. You are allowed to use any parts of the code provided for Lab Exercise 2 at Your Own Time, without the need of acknowledgement.