EAI 320

Practical Assignment 4

Concept by Dr J. Dabrowski
Compiled, edited and reviewed by Johan Langenhoven

28 February 2018

1 Scenario

Tic-Tac-Toe (noughts and crosses) is a zero-sum two player game where each player takes turns marking spaces in a 3×3 grid. One player marks spaces with an 'X', the other player marks spaces with a 'O'. The player who succeeds in placing three of their marks in a horizontal, vertical or diagonal row wins the game. For this game, there are $3^9 = 19683$ possible board layouts (3 possible marks, X, O or empty and 9 possible spaces). There are $9! = 362880$ possible games, which is the size of the game tree, if all possibilities are considered. This is obtained by following the thought process that there are 9 choices for the first mark, 8 for the second, 7 for the third. This leads to $9 \times 8 \times 7... \times 2 \times 1 = 9! = 362880$. This is however just an upper bound, as some of the games are counted twice.

Question 1

a) Write a Python program that implements the alpha-beta search algorithm for any given game. The algorithm should allow for a generic game class to be passed to it. The game class should contain methods as described in section 5 of the prescribed book. The alpha-beta search algorithm should interact with these methods. Typical class features include, but are not limited to:

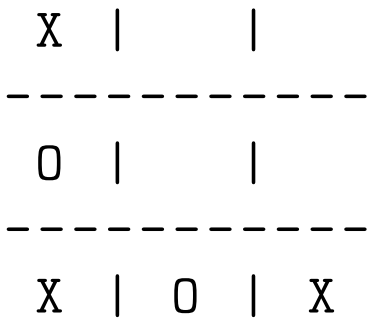
   - The initial board state.
   - A set of operators, or functions, that allow a user to make a move.
   - A function which returns the state of the game i.e Win, Loss, Draw.
   - A pay-off function, which represents a numeric value for the outcome of a game.

b) Create a game class for the Tic-Tac-Toe game. Examples of methods and features that the class should contain are:

   - A board component.
   - Initialiser, which receives a given game state as input.
   - A move function, where a given move is received, checked for validity, and then executed on the board.
   - A function which returns the state of the game i.e Win, Loss, Draw, Unfinished.

Question 2

Consider the game state as illustrated in the image below. Use the software created in Question 1 to find the optimal move for player O. Illustrate the search by drawing the game tree with the nodes that the alpha-beta algorithm evaluates. Show the terminal state values at the leaf nodes. Indicate on the tree which nodes were pruned by the alpha-beta algorithm.



This question's purpose is to make sure your alpha-beta algorithm works as it should. It tests whether your algorithm and the game class work together.

Question 3

Extend the program implemented for Question 1 to create a Tic-Tac-Toe computer game. The game must allow for a human user to play against the AI agent created for Question 1. The user can begin the game by specifying the first move or telling the agent to make the first move. The game is continued until a win, lose or draw results. The user should enter their moves through a command line interface (or graphical user interface if you want to put in the effort). As in question 1, the AI agent must use the alpha-beta algorithm to compute its next move. Document resulting game states for a 'user wins', 'AI wins' and a tie.

Report

You have to write a short technical report for this assignment. Your report must be written in LaTeX. In the report you will give your results as well as provide a discussion on the results. Make sure to follow the guidelines as set out in the separate questions to form a complete report.

Reports will only be handed in as digital copies, but a hard copy plagiarism statement needs to be handed in at the following week's practical session (on the final day of the practical submission).

Deliverable

- Write a technical report on your finding for this assignment.
- Include your code in the digital submission as an appendix.

Instructions

- All reports must be in PDF format and be named report.pdf.
- Place the software in a folder called SOFTWARE and the report in a folder called REPORT.
- Add the folders to a zip-archive and name it `EAI320_prac4_studnr.zip`.
- All reports and simulation software must be e-mailed to **up.eai320@gmail.com** no later than 17:00 on Monday, 19 March 2018. No late submissions will be accepted.
- Use the following format for the subject header for the email: EAI 320 Prac 4 - studnr.
- Bring your plagiarism statements to the practical session on Thursday, 01 March 2018, where they will be collected.
- Submit your report online on ClickUP using the TurnItIn link.

Additional Instructions

- Do not copy! The copier and the copyee (of software and/or documentation) will receive zero for both the software and the documentation.
- For any questions or appointments email me at **up.eai320@gmail.com**.
- Make sure that you discuss the results that are obtained. This is a large part of writing a technical report.

Marking

Your report will be marked as follow:

- 60% will be awarded for the full implementation of the practical and the subsequent results in the report. For partially completed practicals, marks will be awarded as seen fit by the marker. **Commented code allows for easier marking!**
- 40% will be awarded for the overall report. This includes everything from the report structure, grammar and discussion of results. The discussion will be the bulk of the marks awarded.