Task 3 - Lab: Tic-Tac-Toe

Summary

Create a console based (command line interface or CLI) version of the simple game Tic-Tac-Toe in python that supports the development and comparison of computer-controlled AI players (bots).

Your software design can be simple functions (strongly suggested if you are not sure where to start) or a more OO design. The two most important aspects that you must have in your implementation are:

- A clear <u>game loop architecture</u> that contains:
    - input() processing,
    - update()'ing of the game model (including game rules), and
    - render() of the game state to the user.
- A <u>modular block of AI code</u> that can be easily changed.

Stage 1: Create Software Design (Seriously)

Create a design/outline on paper first for the overall game (not the AI). Show your design to the tutor and make sure you are on the right track. Keep your design for use in later work. (It's okay if you use Word or pen/paper for your design.) **Submit this design document to Canvas and commit to your repo.**

If you are unsure of how to approach this, there is an implementation provided. You can have a look at this and modify it, use it for inspiration (ie. cut-n-paste bits if that helps), or ignore it - it's up to you.

Stage 2: Implement Game Loop Model and External AI Call

The input()/update() (or equivalent) blocks of code of your main game loop should support an external call to an AI function/object method that is able to observe the current game state and then make its next move.

Important: Your code must demonstrate the game loop and input/update concepts to be accepted.

Stage 3: Create Two Different AI Bots

Design two different AI's to play the game. Suggested procedure is to write a list of steps (pseudo code) for each AI, and only then implement them. (Strongly suggest that you start by writing code comments for each of the steps in your list, and then "flesh them out".) **Commit your code and your bots to your repo - don't wait until it's perfect.**

For the purpose of quick feedback, **upload the code for your two bots as a <u>single code file</u> to Canvas**. This might be your entire program as a single python file, or just a simple version with the relevant code for the tutor to look at in addition to looking at your repository code.

Extensions: Tic-Tac-Toe AI battles!

Get your own two AI bots to battle it out. Note: you don't need to "render" to screen if your bots are battling - just print the results (and repeat multiple times?!).

Team up with another classmate, modify and share your code to support "AI on AI" competition. (ie. remove the human input and replace with copy-pasted AI code).

By the end of this lab you should have created (or modified the supplied code) the game of Tic-Tac-Toe in python, you should understand the structure and function of a standard game loop, and you should have created at least two different AI bots that can play the game.