ITCS 5153 – Applied AI – Fall 2024
Lab 3

Objectives

- Practice the application of ***Adversarial Search*** algorithms code to **solve game playing problems**.
- Explore the computational complexity implications of using different AI algorithms to solve the same problem.

General Instructions

- You are encouraged to collaborate with other classmates and exchange ideas. However, any **work that you submit needs to be your own**.
- In this assignment, you will write one (1) medium-size *Python* program. The program can be in multiple files or modules, as needed.
- Your code should be organized and commented as needed to make sure that it can be easily understood and maintained.
- Please do not put your name or email anywhere in the code, use only your student ID.
- **Read all instructions carefully**, including any links to external sites (e.g. Wikipedia, Tutorials Point, etc.), *before* you start working on your program.
- Your code needs to work correctly on Python 3.8 to receive full credit, i.e., it does not produce any errors or warnings. *You may comment-out lines that have errors to obtain partial credit for work done.*
- **Test your program thoroughly** to make sure it executes correctly.
- **Submit only your source code**, i.e., the files with the ***.py*** extension.

Part 1: Development Environment Configuration

1. In this assignment, we will use the AI libraries from the **textbook's *Python* code repository** and **Pygame**, both of which you installed and used in earlier assignments. For additional information, see:
    a. https://github.com/aimacode/aima-python/blob/master/README.md
    b. https://www.pygame.org/wiki/GettingStarted

Part 2: Requirements Gathering and Research

1. Read the following Wikipedia entry to familiarize yourself with the rules and gameplay of the ***Connect Four*** game: https://en.wikipedia.org/wiki/Connect_Four.
    a. Make sure that you understand how the game is played.
    b. Consider the game's complexity.
    c. Start thinking of how adversarial search could be used to would build an AI to play this game against a human opponent.

2. Play ***4 in a A Line*** at the following site to get an idea of how the game is played: https://www.mathsisfun.com/games/connect4.html

Part 3: Implementation

Your task is to implement a desktop version of **Connect Four**, which allows users to play against an AI opponent. Your program needs to meet the following requirements:

1. The user interface is built using ***Pygame*** and should allow play on a standard **6x7 board.**
   Hint: Look at the code we used in the pathfinding assignment.
    a. Users will select their move with a simple point-and-click method. The interface needs to prevent invalid moves.
    b. Use ***Alpha-Beta Pruning*** AI algorithm to play against the computer
    c. Users can see the following **information** as the game is played:
        i. Whose turn it is to make the next move.
        ii. The algorithm used to select the move when the AI is playing.
        iii. How long in minutes and seconds the AI has been searching in order to pick the next move.
        iv. A log window (e.g. a textbox with a scroll bar) that is updated every time that the AI makes a move. The log shows the total number of game states (nodes) explored.
    d. When the game is over, a popup dialog or window is shown to indicate who won or whether there was a draw.

    e. There should be buttons to do the following:

   i. Start a **new game**.
   ii. **Restart** the current game.
   iii. **Exit** the program.

2. The AI implementation needs to be based on the ***Python*** code from the **textbook's *Python* code repository**.
   a. You should use as much of the provided code as possible, particularly the search algorithm; however, you can make any changes you deem necessary.
   b. **Use search depth limits (or thresholds) as needed** to ensure that the algorithm is able to complete its search.
   c. You need to **document the changes in the code**, using appropriate comments.

3. Name all your source code files using the following template:
   ***module_name*_123456789.py**
   where *module_name* is replaced with a name representative of the module's functionality (e.g. search) and 123456789 is replaced with your student ID (800#). Please do not use your name, login or email anywhere in your submission.

   Add the following block of comments at the beginning of each file, making sure to replace 123456789 with your student ID (800#) and to include a short description for the module:

```
# UNC Charlotte
# ITCS 5153 - Applied AI - Fall 2024
# Lab 3
# Adversarial Search / Game Playing
# This module implements …
# Student ID: 123456789
```

4. Submit one Zip file containing all your source code files.

Grading Rubric

- This assignment is worth **150 points**. Your work will be graded as follows:
  - 10 points for code that executes in Python 3.8
  - 70 points for code that implements a GUI to play *Connect Four* on a standard 6x7 board.
    - 15 points for the move selection functionality
    - 5 points for functionality to select AI algorithm to play against
    - 5 points for functionality to show whose turn it is to make the next move.
    - 20 points for functionality to show how long in minutes and seconds the AI has been searching in order to pick the next move.
    - 25 points for the log window functionality. This window is updated after every move to show a log of the total number of game states (nodes) explored.

  - 30 points for the code that applies Minimax to select the AI's next move.
  - 40 points for the code that applies Alpha-Beta pruning to select the AI's next move.

- Deductions
  - -5 for meaningless or missing comments.
  - -5 for inconsistent formatting or code that is difficult to read.
  - -5 for incorrectly named files.
  - -5 for files that cannot be unzipped using the standard tools built into MacOS or Windows.
- The lowest possible score is zero, unless you commit an academic integrity violation (e.g. cheating). Please see the syllabus for details.
- Please see the syllabus on *Canvas* for details about the late submission policy.