Programming Project
Knowledge & Reasoning

Overview

The purpose of this formative assignment is for you to deepen your understanding of adversarial search and demonstrate the knowledge and skills required to implement adversarial search in a computer program. This is an individual assignment and while you will not receive individual marks, engaging with the task is important to gain a good understanding of how the underlying algorithm works, which is needed for the final exam assessment. Feedback will be given at the end of the term.

The **default task** outlined in this brief is for you to construct a **Checkers game** (English draughts) in **Python**. The program should be structured in a way that allows pieces to be moved around by a human player and the AI agent/player as per the following basic rules of the checkers game. See the PDF on Canvas with the rules we consider for Checkers.

The information below outlines the features you should try to implement, and some additional features that are not mentioned in the standard rules. Anything else not explicitly specified can be implemented as you see fit. Your program should always **produce legal gameplay** and **have an AI agent** that a human user can play against. Importantly, the design and implementation of your program should be based on an algorithm using **Minimax with Alpha-Beta pruning**.

Requirements

There are nine requirement sections in total. These requirements are split into two main groups: Game Logic and Human-Computer Interface. When you develop your program, try to satisfy as many of these requirements as you can. Some of the requirements are generic for any game; others will only apply to Checkers and can be either ignored or adapted for other games.

Game logic requirements

1. **Gameplay**
    a. Some form of interactive gameplay (Human user vs. computer) with reasonable level of performance (Fluent interactivity, wait times)
    b. Different levels of verifiably effective AI cleverness, adjustable by the user

2. **Search algorithm**
    a. Use of an appropriate state representation
    b. Successor function generates AI moves
    c. AI uses Minimax evaluation with Alpha-Beta pruning
    d. Appropriate use of heuristics

3. **Validation of moves**
    a. No invalid moves are made by the AI
    b. User moves are checked for validity and invalid moves are rejected
    c. Rejection of invalid user moves has clear explanations presented to the user

d. Forced capture - if there is an opportunity to capture an opponent piece, then a capturing move must be made by the current player. If there is more than one capturing opportunity at the same time, the player may choose which one to take

4. **Other features**
    a. Multi-leg capturing moves for the user
    b. Multi-leg capturing moves for the AI
    c. King conversion at baseline (the king's row) as per the normal rules
    d. Regicide - if a normal piece manages to capture a king, that piece is instantly crowned king and then the current turn ends
    e. A help facility that provides hints about available moves, given the current game state, or even employ the AI to make suggestions for the human player
    f. AI vs AI game play option/mode

Human-Computer interface requirements

5. Some kind of game state / board representation displayed on screen
6. The interface properly updates the display after completed moves (User and AI moves)
7. A simple and intuitive text-based non-graphical interface to play the game must be provided
8. The interface pauses appropriately to show the AI moves, esp. if there are several in a row
9. Some display of the rules, or information on how to play the game should be provided

Very short report

We would like every submission to be accompanied by a short report of about 3-4 pages in length, including title page. This report will be useful for you to practice verbalising your understanding of the AI functionality that you have implemented. There is no a hard-word limit given, but as a guideline, try not to exceed 1200 words in total (excluding Appendices, if needed).

Your report should have a **title page** with your candidate number (not your name!) and contain the following sections:

1. **Introduction** – No more than a page (include a screen snapshot of your program's user interface). Describe your general approach to solving the problem. For example, you could write a few sentences about your design, including a list the functions/methods used in your program, what they do, and any relevant information about their functionality.
2. **Description of the AI functionality** – Use this part of the report to explain 1) how your Minimax/Alpha-Beta algorithm works specifically for your implementation (omit any general textbook-like description – your readers will know how the algorithm works in principle). Describe, in particular, 2) how you make use of heuristics and what kind of information is used to make them work, and 3) your design for controlling the AI difficulty of the game.
3. **Appendix** (Source code, perhaps with line numbers for easier reference from your text).

Programming Project
Knowledge & Reasoning

Submission Requirements

Submit your coursework online on **Canvas** as a **single ZIP file** that contains all of the following:

1. **Your program files** (e.g., your project folder including source code and assets used)
2. **Clear and detailed instructions on how to run your program** (readme.txt file)
3. **Your report in PDF**

Important note about formatting and submission. Name your main ZIP file using the following format: KR2025-XXXXXXXX.zip where XXXXXXXX is your unique university candidate number

Avoid academic misconduct

Please keep in mind that this is an individual formative assessment task (so, marks are not awarded with the development of this coursework submission); the related University rules on academic integrity apply. For example, it would not be acceptable to submit programs that were developed by other people. If you are finding it necessary to reuse bits of code off the Web, Github, etc. that are not of your own design, then, please add references to the original source(s) and indicate in the introduction the parts that are your own work. If you use Generative AI tools in any parts of your code, please, indicate that explicitly. The short report is a personal manuscript, and for that Generative AI tools are not allowed.

Submission Deadline

Week 7: Sunday, 16th of November, by 23:59

Good luck!