

Alpha-Beta Pruned Chess AI Engine

Project Description

This project involves the development of an interactive chess game in which a human player competes against an AI agent. The AI engine must implement the **Alpha-Beta pruning algorithm**, an optimized version of the minimax algorithm widely used in adversarial search problems.

The project emphasizes understanding core AI techniques for decision-making in competitive environments, especially two-player games. It also offers flexibility for students to explore further enhancements such as **move ordering**, **heuristic evaluation functions**, and other performance optimizations.

Students are expected to implement a functional chessboard that enforces the official rules of chess, including legal move validation, check, and checkmate detection. The AI component should be capable of planning multiple moves ahead using alpha-beta pruning and should adapt to different configurations.

Project goals

The goals of this project are two-fold:

1. **Functional Chess Game:** Build a complete and playable chess interface where a human player can make any legal move. The game must enforce check, checkmate, and no illegal moves should be allowed.
2. **AI Chess Agent using Alpha-Beta Pruning:**
 - Implement an AI agent that uses alpha-beta pruning to search for **at least 4 plies** (2 moves per player) into the future.
 - The agent should **consistently outperform a random move generator**.
 - The alpha-beta engine should be **configurable**, allowing experiments with different depths, evaluation strategies, or enhancements (optionally) (e.g., move ordering, iterative deepening).
 - It must be possible to simulate a match between two AI agents using different alpha-beta configurations, with the stronger agent winning in most simulations.

Assumptions

Students are allowed and encouraged to make and document reasonable assumptions to guide their implementations. Some examples include:

- The choice of **heuristic evaluation function** to score non-terminal board states.
- How **move ordering** is implemented (e.g., ordering captures first to improve pruning efficiency).
- Handling of special rules like **castling**, **en passant**, and **pawn promotion**—either implemented or excluded, but clearly stated.
- The internal **representation of the board** and **search tree**.
- Stating optimizations if they are used with the motivation behind each optimization.
- Programming language and libraries (e.g., using pygame for GUI or python-chess for board validation).

Milestones

Milestone 1: Functional Chessboard + Random Agent (10 marks)

- Develop the complete chessboard interface with rule enforcement (legal move generation, check, and checkmate detection). (5 marks)
- Implement a **random move generator** agent as the initial AI opponent. (3 marks)
- Human player should be able to play against the random agent interactively. (2 marks)

Milestone 2: Alpha-Beta AI Engine + Simulation Experiments (20 marks)

- Implement the **alpha-beta pruning algorithm** as the main AI opponent. (10 marks)
- Enable **depth configuration**, move ordering, and optional enhancements. (3 marks)
- Evaluate and report on performance and outcomes. (3 marks)
- Simulate games between:
 - Alpha-beta agent vs random agent (alpha-beta must win most games). (2 marks)
 - Alpha-beta agent (simple config) vs alpha-beta agent (optimized config). (2 marks)

Deliverables

1. **Source Code:**

- Fully functional codebase that runs the chess game with both random and alpha-beta agents.
- Code must be modular and readable, with clear documentation on how to configure AI parameters.

2. **Documentation:**

A short report detailing:

- Project overview
- Assumptions made
- Implementation choices
- Description of the AI agent(s) and the strategy made
- Heuristic function motivation
- detailed problem formulation and task environment
- Parameter configurations and experimental setup
- Results and observations from simulations

3. **Demonstration Meeting:**

Each student/team will attend a brief live session to:

- Run their program
- Show a few moves between human vs AI and AI vs AI
- Discuss their approach and results