

# CS747 Assignment 3

Geet Sethi (23B2258)

November 2, 2025

## 1 Overview

All non-random agents evaluate positions with a scalar score from the perspective of the player who started the search ('original\_turn'). Positive is good for that player, negative is bad.

Material values used are:

- Pawn: 100
- Knight: 300
- Bishop: 300
- Rook: 500
- Queen: 900
- King: 0 (since not having the king is not a valid game state as the game will end immediately on losing the king)

We also have the following constants:

- Checkmate score: 1000000
- Draw score: 0
- Check bonus: 40
- Pin bonus: 30
- Mobility bonus: 2

The use of these constants is explained in the sections below.

## 2 Task1Agent

**Approach for selecting moves:** We use a simple MiniMax algorithm with search up to depth 2. For each legal move on the current position, we make the move on a copy of the board, recursively run `minimax` to remaining depth with alternating maximizing/minimizing roles, and pick the move with the highest returned score.

**Heuristics/evaluation:** We use the following heuristics to evaluate the board:

- **Material:** sum of piece values for both sides as defined previously.
- **Pawn advancement bonus:** bonus proportional to progress toward promotion (encourages pawns to promote).
- **Check bonus:** bonus when delivering check against the opponent's king.

Note that we always evaluate the board from the perspective of the original turn player since this is handled appropriately in the minimax function. So we always add the score of the original turn player to the score of the board and subtract the score of the other player from the score of the board to get the final evaluation score of the board from the perspective of the original turn player to judge the move quality.

**Key observations:** Material-only play was not giving any incentive to win the game by taking out the opponent's king due to which all the games were resulting in a stalemate and hence a tie. So adding a small check bonus to the evaluation function helped in winning the games by taking out the opponent's king. But this too was not always enough since at the start there won't be any moves to get a check on the opponent's king and hence the game will end in a stalemate. So we also add a pawn advancement bonus to the evaluation function to encourage pawns to go to the other side of the board and promote.

### 3 Task2Agent

**Approach for selecting moves:** We use a MiniMax algorithm with alpha-beta pruning. We search up to a depth of 3. For each legal move on the current position, we make the move on a copy of the board, recursively run `minimax` to remaining depth with alternating maximizing/minimizing roles, and pick the move with the highest returned score. We also use a cache to store the best move and score for a given position and depth so as to avoid re-searching the same position at the same depth again and speed up the search.

We also shuffle the moves for each position to get a different game sequence and potentially better pruning if the move ordering is poor.

**Heuristics/evaluation:**

- **Material** (same as in Task 1).
- **Check bonus:** (same as in Task 1)
- **Piece-square tables:** We give a bonus to more powerful positions for each type of piece according to its abilities (for example, pawns are encouraged to advance; minor/major pieces are encouraged to control the center; king prefers to stay on the back rank for safety).
- **Pinned piece penalty:** We subtract a penalty per pinned piece (since being pinned reduces flexibility of the piece).

**Key observations:** The piece-square tables significantly improved move quality (especially centralization and safe king posture on the small board). However, the increased depth of 3 plies made the search take a lot of time and was not finishing within the time limits. So we added a cache to store the best move and score for a given position and depth so as to avoid re-searching the same position at the same depth again and speed up the search.

## 4 Task3Agent

**Approach for selecting moves:** Same search framework as Task 2 with an augmented evaluation that includes mobility and an increase in search depth to 4 plies.

**Heuristics/evaluation:**

- All Task 2 terms (material, piece-square tables, check bonus, pinned penalty).
- **Mobility:** adds a small bonus proportional to the number of legal moves (computed by toggling the turn and recomputing the valid moves). This rewards flexible and non-cramped positions.

**Key observations:** While mobility improved search guidance in quiet positions, the main gain in the score was from the increased search depth to 4 plies.

## 5 Results

Agent	Opponent	Total Score	Draws	Avg. Time (s)	Avg. Game Length
Task1Agent	RandomAgent	38	62	0.0009	11.0
Task2Agent	RationalAgent	29	61	0.0172	16.0
Task3Agent	RationalAgent	75	21	0.0603	12.0
Task4Agent	RationalAgent	87	13	0.3742	15.0