Department of Computer Science and Engineering

Course Code: CSE-3636
Course Title: Artificial Intelligence Lab
Project Title: AI based Chess Game using Python

Project Documentation

Week 10: Nega Max & Alpha Beta Pruning

Submitted by

Lamisa Mashiat (C201249)

Sohana Tasneem (C201266)

Sadia Haque Chowdhury (C201270)

Submitted to

Subrina Akter
Assistant Professor
Dept. of CSE

20th May, 2023

Introduction: This is an updated version of the ChessAI.py script that implements the NegaMax algorithm with alpha-beta pruning to improve the efficiency of the chess AI. The NegaMax algorithm is a variant of the Minimax algorithm that simplifies the code and improves the performance. The alpha-beta pruning optimizes the search by eliminating unnecessary branches.

Functionality: The script contains two main functions - findBestMove and findMoveNegaMaxAlphaBeta.

findBestMove(): This function takes the current game state and a list of valid moves and returns the best move for the current player. The function uses the findMoveNegaMaxAlphaBeta function to calculate the best move with alpha-beta pruning. The function shuffles the list of valid moves to randomize the search and reduce predictability.

findMoveNegaMaxAlphaBeta(): This function is a recursive function that searches for the best move using the NegaMax algorithm with alpha-beta pruning. The function takes the current game state, a list of valid moves, depth, alpha, beta, and turnMultiplier. The depth parameter is the maximum depth of the search tree, and the alpha and beta parameters are used for alpha-beta pruning. The turnMultiplier is used to calculate the score based on the player's turn.

The function iterates over the valid moves and calculates the score for each move recursively using the NegaMax algorithm. If the score is higher than the current maxScore, it updates the maxScore and stores the move in the nextMove variable. If the current depth is equal to the maximum depth, it stores the move in the nextMove variable.

The function implements alpha-beta pruning to optimize the search by eliminating unnecessary branches. If the maxScore is greater than the alpha value, it updates the alpha value. If the alpha value is greater than or equal to the beta value, it breaks out of the loop and returns the maxScore.

Variables: The script uses several variables to store the game state, the score for each piece, the values for checkmate and stalemate, and the maximum depth of the search tree.

pieceScore: A dictionary that stores the value of each piece based on its type.

CHECKMATE: A constant variable that stores the value for a checkmate.

STALEMATE: A constant variable that stores the value for a stalemate.

DEPTH: A constant variable that stores the maximum depth of the search tree.

nextMove: A global variable that stores the best move for the current player.

Conclusion: This updated version of the ChessAI.py script implements the NegaMax algorithm with alpha-beta pruning to improve the efficiency of the chess AI. The NegaMax algorithm simplifies the code and improves the performance, and the alpha-beta pruning optimizes the search by eliminating unnecessary branches. The script uses several variables to store the game state, the score for each piece, the values for checkmate and stalemate, and the maximum depth of the search tree.