Jayam Sutariya

Dr. Nguyen

CS 482

3rd October 2021

Project 1 Report

Introduction

For this first project, we were required to implement one the Artificial Intelligence concepts we learned in class. The concept is that of the Minimax Search Algorithm. The algorithm is very useful in 2 player games. One such application would be Tic Tac Toe. With our knowledge on the Minimax Search algorithm combined with some basic programming principles such as recursion is to create a program that finds the optimal move in the game of tic-tac-toe.

Explanation of approach

The API for the program was already established. The program loads the tic-tac-toe board file, and after finding the optimal move, saves the board to a file. The essence of the remainder of the program remained in the finding the optimal move. This was done by the use of the Minimax Search Algorithm. Based on the pseudocode provided in the lecture slides, I wrote a similar version of the algorithm that worked with established API.

The API already calculated which user's turn it is. Based on that, the program would go through every empty square and find a score for each empty square by going through all subsequent optimal solutions for alternating players until the game either results in a draw or results in a win by either one of the players. Whichever empty square results in the highest score is marked as the player's square. This whole process is clearly recursive.

The program only provides the best move based on the current state of the tic-tac-toe board. To continue the game, further command line arguments need to be executed to see further optimal moves. After a certain number of moves, the game will either end based on the state of the board, or it will come to tie. Nonetheless, the program will not allow you to continue if the board is full. Moreover, at the end of the game, the program either declares the winner of the game, or declares the game a tie.

Results and Discussion

There are a lot of scenarios that can be played in a game of tic-tac-toe. However, all the scenarios can be classified into one of three categories based on the starting state of the board. The three categories being win for player 1, win for player 2, or a tie. Multiple scenarios were tested for three categories and the results came out as expected. Scenarios where the game was clearly a tie, the program gave results that produced a tie. Scenarios where either one of the players should be winning, the program again gave the optimal moves to a victory for the player in a winning situation. The scenarios were tested with fewest number of moves already played, the results stayed accurate.