CS580: INTRODUCTION TO ARTIFICIAL INTELLIGENCE

PROJECT – 2

G01369862

Introduction:

Adversarial search is a way that artificial intelligence finds the best moves for a player in a game where there's an opponent trying to win too. The goal is for the AI to increase its chances of winning while also considering what the opponent might do. For our project, we're going to use two popular techniques in adversarial search, the Minimax algorithm and Alpha-Beta pruning, to play a modified version of Connect 4. We want to create an AI that can play the game really well against a human player and see how it does under different conditions. The report we're putting together is split into five parts: Introduction, Background, Proposed Approach, Experimental Results, and Conclusions.

Background:

The Minimax algorithm is a way to make decisions and figure out the best move for a player in game theory. It's all about finding the best move even when there's an opponent who's trying to do the same thing. Here's how it works: the algorithm checks out all the possible moves from where the game is currently at and gives each one a score based on what might happen. The MAX player is trying to get the highest score possible, while the MIN player wants the lowest score. The algorithm assumes that both players are playing to the best of their abilities, so they're always making the smartest move they can based on the situation.

Alpha-Beta pruning is a way to make the Minimax algorithm work even faster. It's all about eliminating branches of the search tree that are worse than other branches that we've already looked at. By doing this, we don't need to look at as many nodes, which speeds everything up and makes it more efficient.

Connect 4 is a game that two people can play against each other. The game board is a grid that's six squares tall and seven squares wide. Each player takes turns dropping a disc of their color into one of the seven columns. The game ends when one player gets four of their discs in a row, either up and down, side to side, or diagonally. Whoever manages to get four in a row first is the winner of the game**.**

Proposed Approach:

To solve the Connect 4 game, we're going to use the Minimax algorithm along with Alpha-Beta pruning. This algorithm will check out all the different moves that we could make from where we're at in the game, and then score each one based on how things could turn out. To make sure we're not looking at too much stuff, we'll assign points to the depth of the algorithm so that we only search as deep as we need to. We're also going to make a menu that lets people choose things like what color the board should be, what the player's name is, who gets to go first, and how deep the algorithm should search.

Experimental Results:

We want to see how well our agent is doing, so we're going to play 35 games against it under different conditions. We'll play each of the seven columns with each of the five levels of depth, so we'll end up with 35 games in total. For each game, we'll keep track of things like who won, how many moves it took to win, and anything else that seems important. And just to make everything easier to understand, we'll also include a picture like a bar graph or scatter plot that helps show what's going on with the data.

Conclusions:

After looking at the results of our experiment, we'll be able to figure out the best level of depth for the Minimax algorithm to search for. This will help our agent win more often. We'll also come to some other important conclusions based on what we found. And once we're all done, we'll take some time to think about what we did well, what could've been better, and what we might do differently in the future.