

Courtney Wood

Artificial Intelligence Assignment #1 Report

I used a minimax algorithm with alpha beta pruning to create my player. The player is efficient and makes all moves in under 10 seconds.

When the chooseMove() function is initially entered, I create an ArrayList of all "next" moves on the board, which are Move objects with row and col set to the location of open tiles on the board that neighbor occupied spaces. I chose to only consider these tiles as moves instead of all open tiles because tiles that are not around occupied spaces are not optimal or relevant to the game. Using this ArrayList, I create a new ArrayList of boards that have each move in the arraylist placed on a board. I then call minimax on all these boards. Initially, I discovered that this method was slow and would often time out, especially if one player was about to win. I decided that before entering minimax, I could add a check to see if the current child board was a winning board, and if it was, to immediately make that move, instead of discovering it through minimax. I also created checks for if the opponent was about to win, where it would immediately block that move instead of discovering the move through minimax. This significantly reduced the time of my minimax algorithm.

My minimax algorithm is fairly straightforward. It has a depth of 3, because this was the depth that allowed it to be the smartest while retaining its speed. The base cases are when one player wins, or when the depth is 0. When the depth reaches 0, I calculate the score using determineScore(). My determineScore method looks for lines of 2, 3 and 4 and assigns scores accordingly for each player. The player receives positive values that increase as the length of the chain increases, and the opponent receives negative values. These are summed and the sum is the score of the board.

The function I use to check the number of tiles in a line is my checkForNumInARow() function. It takes in the starting row and column and the direction, and returns true if there is a line in that direction. I use this function to check to see if a board is a winning board, and to calculate the score of a board.