Assignment 4

ARTIFICIAL INTELLIGENCE

Name: M. Nandith Reddy

UIN: 01066678

E mail: [nmala001@odu.edu](mailto:nmala001@odu.edu)

Demonstration:

The Connect 4 game is developed using the minimax algorithm.  
The Game is played between the Player(User) and the Computer AI.

The one who connects four in a row is a winner.

The human move is represented by “X” and the Computer move is represented by “O”

Step 1: Human makes a move in the column No. 1

Step 2: Computer makes a move in the column No.7

Step 3: Human makes a move in the column No. 3

Step 4: Computer makes a move in the column No.2

Step 5: Human makes a move in the column No. 5

Step 6: Computer makes a move in the column No.4

Step 7: Human makes a move in the column No. 6

Step 8: Computer makes a move in the column No.7

Step 9: Human makes a move in the column No. 7

Step 10: Computer makes a move in the column No.6

Step 11: Human makes a move in the column No. 5

Step 12: Computer makes a move in the column No.5

Step 13: Human makes a move in the column No. 5

Step 14: Computer makes a move in the column No.4

Step 15: Human makes a move in the column No. 6

Step 16: Computer makes a move in the column No.4

Step 17: Human makes a move in the column No. 4

Step 18: Computer makes a move in the column No.6

Step 19: Human makes a move in the column No. 6

Step 20: Computer makes a move in the column No.7

Step 21: Human makes a move in the column No. 6

Step 22: Computer makes a move in the column No.7

Step 23: Human makes a move in the column No. 4

Step 24: Computer makes a move in the column No.7

Search Algorithm:

My initial implementation used solely the Minimax search algorithm and could search in depth of six for around one minute. Obviously this was not fast enough, my first improvement was implementing Alpha Beta Pruning to minimize the search space. In order to improve the pruning I changed the order in which moves are considered, stating in the middle and alternating one row left and right. This resulted in tremendous speed up of the search. However, searching only to fixed depth it was hard to control the time for search to be in the given time limit.

Evaluation Function:

The AI engine is based off the minimax algorithm. Strictly speaking, the computer first looks for a winning move. If no winning move is found, it blocks the opponent's winning move, if one exists. If the opponent has no winning move, it runs the minimax algorithm up to the designated depth.

Solution:

Connect-4!

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

Human 'X' moves.

Give column (1-7): 1

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| X | - | - | - | - | - | - |

AI 'O' moves.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| X | - | - | - | - | - | O |

Human 'X' moves.

Give column (1-7): 3

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| X | - | X | - | - | - | O |

AI 'O' moves.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| X | O | X | - | - | - | O |

Human 'X' moves.

Give column (1-7): 5

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| X | O | X | - | X | - | O |

AI 'O' moves.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| X | O | X | O | X | - | O |

Human 'X' moves.

Give column (1-7): 6

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| X | O | X | O | X | X | O |

AI 'O' moves.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | O |

| X | O | X | O | X | X | O |

Human 'X' moves.

Give column (1-7): 7

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | X |

| - | - | - | - | - | - | O |

| X | O | X | O | X | X | O |

AI 'O' moves.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | X |

| - | - | - | - | - | O | O |

| X | O | X | O | X | X | O |

Human 'X' moves.

Give column (1-7): 5

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | X |

| - | - | - | - | X | O | O |

| X | O | X | O | X | X | O |

AI 'O' moves.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | O | - | X |

| - | - | - | - | X | O | O |

| X | O | X | O | X | X | O |

Human 'X' moves.

Give column (1-7): 5

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | X | - | - |

| - | - | - | - | O | - | X |

| - | - | - | - | X | O | O |

| X | O | X | O | X | X | O |

AI 'O' moves.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | X | - | - |

| - | - | - | - | O | - | X |

| - | - | - | O | X | O | O |

| X | O | X | O | X | X | O |

Human 'X' moves.

Give column (1-7): 6

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | X | - | - |

| - | - | - | - | O | X | X |

| - | - | - | O | X | O | O |

| X | O | X | O | X | X | O |

AI 'O' moves.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | - | X | - | - |

| - | - | - | O | O | X | X |

| - | - | - | O | X | O | O |

| X | O | X | O | X | X | O |

Human 'X' moves.

Give column (1-7): 4

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | X | X | - | - |

| - | - | - | O | O | X | X |

| - | - | - | O | X | O | O |

| X | O | X | O | X | X | O |

AI 'O' moves.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | - | - |

| - | - | - | X | X | O | - |

| - | - | - | O | O | X | X |

| - | - | - | O | X | O | O |

| X | O | X | O | X | X | O |

Human 'X' moves.

Give column (1-7):

6

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | X | - |

| - | - | - | X | X | O | - |

| - | - | - | O | O | X | X |

| - | - | - | O | X | O | O |

| X | O | X | O | X | X | O |

AI 'O' moves.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | - | - | X | - |

| - | - | - | X | X | O | O |

| - | - | - | O | O | X | X |

| - | - | - | O | X | O | O |

| X | O | X | O | X | X | O |

Human 'X' moves.

Give column (1-7): 4

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | X | - | X | - |

| - | - | - | X | X | O | O |

| - | - | - | O | O | X | X |

| - | - | - | O | X | O | O |

| X | O | X | O | X | X | O |

AI 'O' moves.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| - | - | - | - | - | - | - |

| - | - | - | X | - | X | O |

| - | - | - | X | X | O | O |

| - | - | - | O | O | X | X |

| - | - | - | O | X | O | O |

| X | O | X | O | X | X | O |

AI computer 'O' wins!

Game over.

Analysis:

The program is developed using MiniMax . A MiniMax tree is used in AI systems to take a board and determine which move will be most advantageous. It uses a heuristic to assign a value to a move based on the principle of minimizing the potential to lose and maximizing the potential to win. Each tree has a depth that represents the number of moves in advance that it uses to evaluate the current move. We designed our MinMax tree implementation in a way which allowed us to simply plug in new heuristic evaluation functions as we developed them. We developed and implemented the following heuristics:

● First: Ranks a board based on whether or not the board is a win, loss, or an immediate set-up for an opponent victory.

● Second: First Heuristic + takes into account the depth of the board within the MinMax tree

● Third: Second Heuristic + takes into account the number of series of three consecutive tokens

● Fourth: Third Heuristic + considers the depths of the series of three tokens

● Fifth: Fourth Heuristic + gives value to the number of wins a board has

● Sixth: Fifth Heuristic + looks for a situation in which a win is guaranteed, where a threat has two adjacent, win-inducing available slots. A threat is a series of three consecutive tokens which could potentially result in a win either immediately or later in the game

● Seventh: Sixth Heuristic + looks for a situation in which two consecutive tokens have both adjacent slots open, in order to possibly set up for an automatic victory

● Eighth: Seventh Heuristic + subtracts the value given for the opponent with heuristics 4, 5, 6, and 7 in order to judge the other player’s performance with a board