**CS4750-HW4 Report**

**Team member’s names:** Song Vu Nguyen, Caleb Fagan, Charlermpon Thongmotai

**References:**

**Description**: Implement the minimax algorithm in Python to play a two-player, four-in-a-row game, which is a variation of tic-tac-toe: two players, X and O, take turns marking the spaces in a 6×6 grid. Player 1 The player who succeeds in placing 4 of their marks consecutively in a horizontal, vertical, or diagonal row wins the game. The heuristic function that we used for our minimax algorithm is h(n) = 5\*[# of two-side-open-3-in-a-row for me]–10\*[# of two-side-open-3-in-a-row for opponent] + 3\*[# of one-side-open-3-in-a-row for me]–6\*[# of one-side-open-3-in-a-row for opponent] + [#of open-2-in-a-row for me] - [# of open-2-in-a-row for opponent] where:

“one-side-open-3-in-a-row”: there is a blank space at one end of a 3-in-a-row.

“two-side-open-3-in-a-row”: there are blank spaces at both ends of a 3-in-a-row.

“open-2-in-a-row”: there are blank spaces at one or both ends of a 2-in-a-row.

Player 1, with symbol X, running the minimax algorithm searching in the depth of 4 moves ahead. Player 2, with symbol O, running the minimax algorithm searching in the depth of 2 moves ahead. Player 1 moves first by placing X in the middle of the board. Each player then take turn placing their symbol on the board in the location that give them the highest heuristic value after running the minimax algorithm. The result of the game is listed below

**Result:**

1st Move:

* [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

2nd Move:

* ['O' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

3rd Move:

* ['O' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' 'X' ' ' ' ' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

4th Move:

* ['O' 'O' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' 'X' ' ' ' ' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

5th Move:

* ['O' 'O' 'X' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' 'X' ' ' ' ' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

6th Move:

* ['O' 'O' 'X' 'O' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' 'X' ' ' ' ' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

7th Move:

* ['O' 'O' 'X' 'O' ' ' ' ']   
  [' ' 'X' ' ' ' ' ' ' ' ']   
  [' ' ' ' 'X' ' ' ' ' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

8th Move:

* ['O' 'O' 'X' 'O' 'O' ' ']   
  [' ' 'X' ' ' ' ' ' ' ' ']   
  [' ' ' ' 'X' ' ' ' ' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

9th Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  [' ' 'X' ' ' ' ' ' ' ' ']   
  [' ' ' ' 'X' ' ' ' ' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

10th Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' ' ' ' ' ' ' ' ']   
  [' ' ' ' 'X' ' ' ' ' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

11th Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' 'X' ' ' ' ' ' ']   
  [' ' ' ' 'X' ' ' ' ' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

12th Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' 'X' 'O' ' ' ' ']   
  [' ' ' ' 'X' ' ' ' ' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

13th Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' 'X' 'O' 'X' ' ']   
  [' ' ' ' 'X' ' ' ' ' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

14th Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' 'X' 'O' 'X' 'O']   
  [' ' ' ' 'X' ' ' ' ' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

15th Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' 'X' 'O' 'X' 'O']   
  ['X' ' ' 'X' ' ' ' ' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

16th Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' 'X' 'O' 'X' 'O']   
  ['X' 'O' 'X' ' ' ' ' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

17th Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' 'X' 'O' 'X' 'O']   
  ['X' 'O' 'X' 'X' ' ' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

18th Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' 'X' 'O' 'X' 'O']   
  ['X' 'O' 'X' 'X' 'O' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

19th Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' 'X' 'O' 'X' 'O']   
  ['X' 'O' 'X' 'X' 'O' ' ']   
  [' ' 'X' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

20th Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' 'X' 'O' 'X' 'O']   
  ['X' 'O' 'X' 'X' 'O' 'O']   
  [' ' 'X' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

21st Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' 'X' 'O' 'X' 'O']   
  ['X' 'O' 'X' 'X' 'O' 'O']   
  ['X' 'X' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

22nd Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' 'X' 'O' 'X' 'O']   
  ['X' 'O' 'X' 'X' 'O' 'O']   
  ['X' 'X' 'O' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

23rd Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' 'X' 'O' 'X' 'O']   
  ['X' 'O' 'X' 'X' 'O' 'O']   
  ['X' 'X' 'O' 'X' ' ' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

24th Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' 'X' 'O' 'X' 'O']   
  ['X' 'O' 'X' 'X' 'O' 'O']   
  ['X' 'X' 'O' 'X' 'O' ' ']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

25th Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' 'X' 'O' 'X' 'O']   
  ['X' 'O' 'X' 'X' 'O' 'O']   
  ['X' 'X' 'O' 'X' 'O' 'X']   
  [' ' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

26th Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' 'X' 'O' 'X' 'O']   
  ['X' 'O' 'X' 'X' 'O' 'O']   
  ['X' 'X' 'O' 'X' 'O' 'X']   
  ['O' ' ' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

27th Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' 'X' 'O' 'X' 'O']   
  ['X' 'O' 'X' 'X' 'O' 'O']   
  ['X' 'X' 'O' 'X' 'O' 'X']   
  ['O' 'X' ' ' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

28th Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' 'X' 'O' 'X' 'O']   
  ['X' 'O' 'X' 'X' 'O' 'O']   
  ['X' 'X' 'O' 'X' 'O' 'X']   
  ['O' 'X' 'O' 'X' ' ' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time:

29th Move:

* ['O' 'O' 'X' 'O' 'O' 'X']   
  ['O' 'X' 'X' 'O' 'X' 'O']   
  ['X' 'O' 'X' 'X' 'O' 'O']   
  ['X' 'X' 'O' 'X' 'O' 'X']   
  ['O' 'X' 'O' 'X' 'X' ' ']   
  [' ' ' ' ' ' ' ' ' ' ' ']
* # of nodes generated:
* CPU execution time: