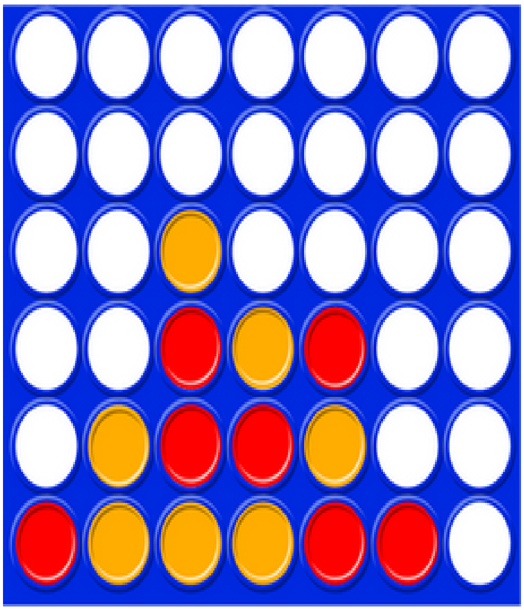
Introduction To Artificial Intelligence Project (A.i)

Connect 4 game

name	ID	Department
Abdelrahman Ahmed Fathy Abdelaziz	20221441784	A.i
Ahmed Mostafa Abdelrahman	20221372883	A.i
Abdelrahman Tarek zaki	20221442265	A.i
Mazen Gaber	20221372110	A.i



Creating a Connect Four Game in Java

—{Project Description }—

➔Connect 4 is a two-player game dropping their colored discs from the top into a grid. The pieces fall straight down, occupying the next available space within the column.

➔The objective :-of the game is to connect-four of one's own discs of the same color next to each other vertically, horizontally, or diagonally. The two players keep playing until the board is full. The winner is the player having greater number of connected-fours.

—{Project ToDo list }—

1) : with mode human vs computer the user choose whether to use alpha-beta pruning or not

2) The game dimensions are as follows (width ≥ 7, length ≥ 6).

—{Project Algorithm }—

1) • Minimax without alpha-beta pruning

2)• Minimax with alpha-beta pruning

➔Heuristic Pruning🡐

Minimax without alpha-beta pruning pseudocode:-

```
minimax(node, depth, maximizingPlayer)

    if depth = 0 or node is a terminal node
        return the heuristic value of node

    if maximizingPlayer
        bestValue := −∞
        for each child of node
            v := minimax(child, depth − 1, FALSE)
            bestValue := max(bestValue, v)
       return bestValue

    else    (* minimizing player *)
        bestValue := +∞
        for each child of node
            v := minimax(child, depth − 1, TRUE)
            bestValue := min(bestValue, v)
        return bestValue
```

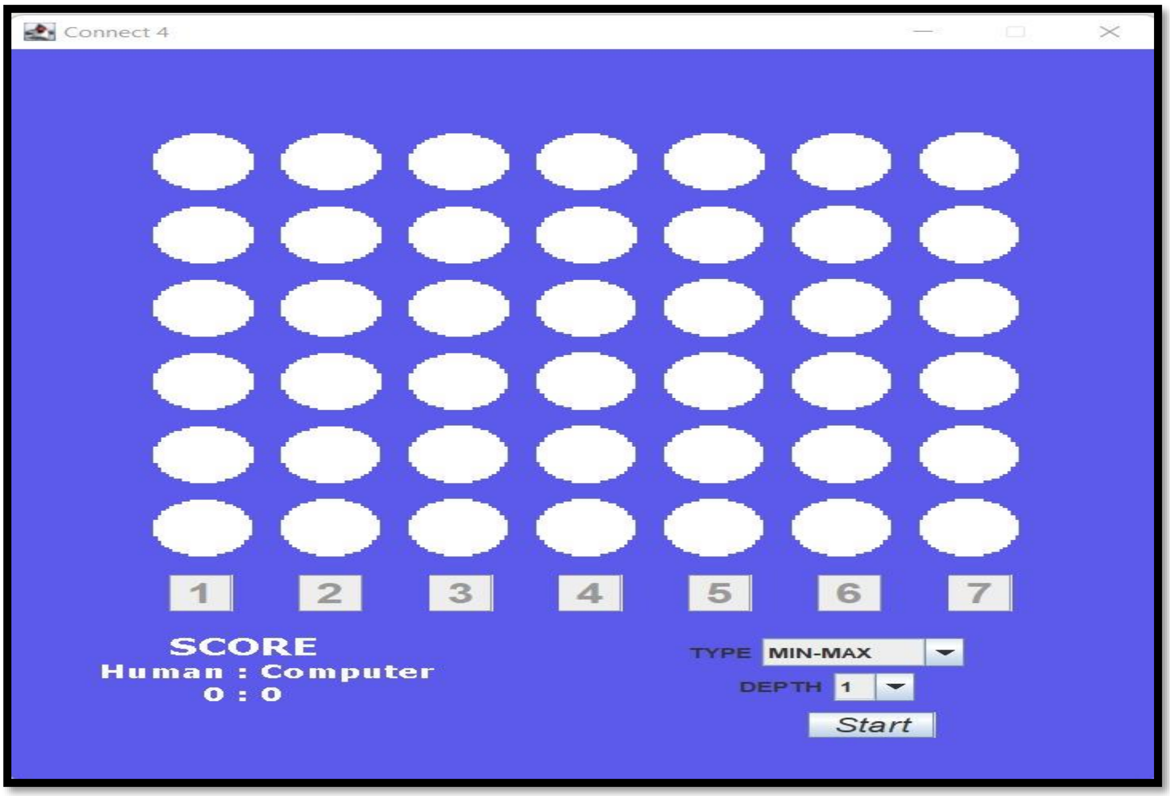
Minimax with alpha-beta pruning pseudocode:-

```
function minimax (node, depth, isMaximizingPlayer, alpha, beta):
if node is a leaf node:
return value of the nod
      if MaximizingPlayer:
     maxValue = -INFINITY
     for each child node:
value = minimax(node, depth+1, false, alpha, beta)
maxValue = max( maxValue, value)
     alpha = max( alpha, bestVal)
     if beta <= alpha:
    cut off
    return maxValue
else :
minValues = +INFINITY
for each min nodes:
value = minimax(node, depth+1, true, alpha, beta)
minValues = min( minValues, value)
beta = min(beta, minValue)
if beta <= alpha:
cut off
return minValue
```

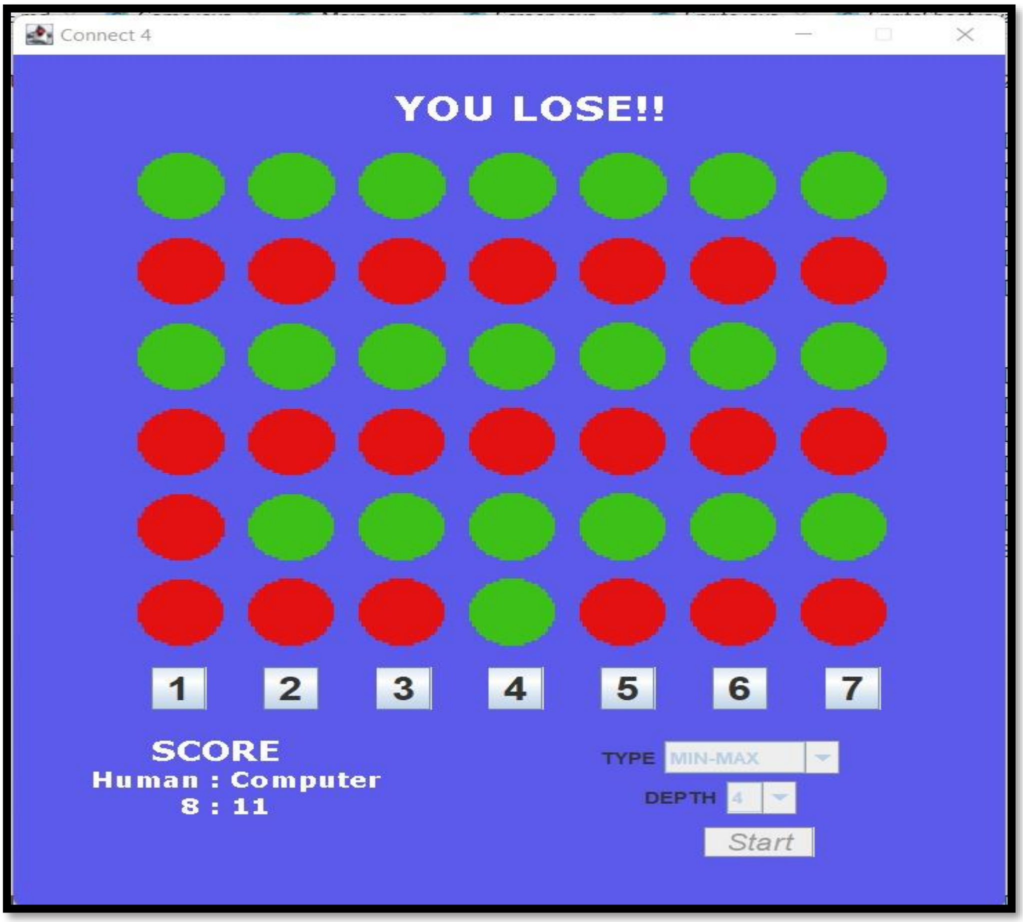
Minimax without alpha-beta pruning

Sample run Test

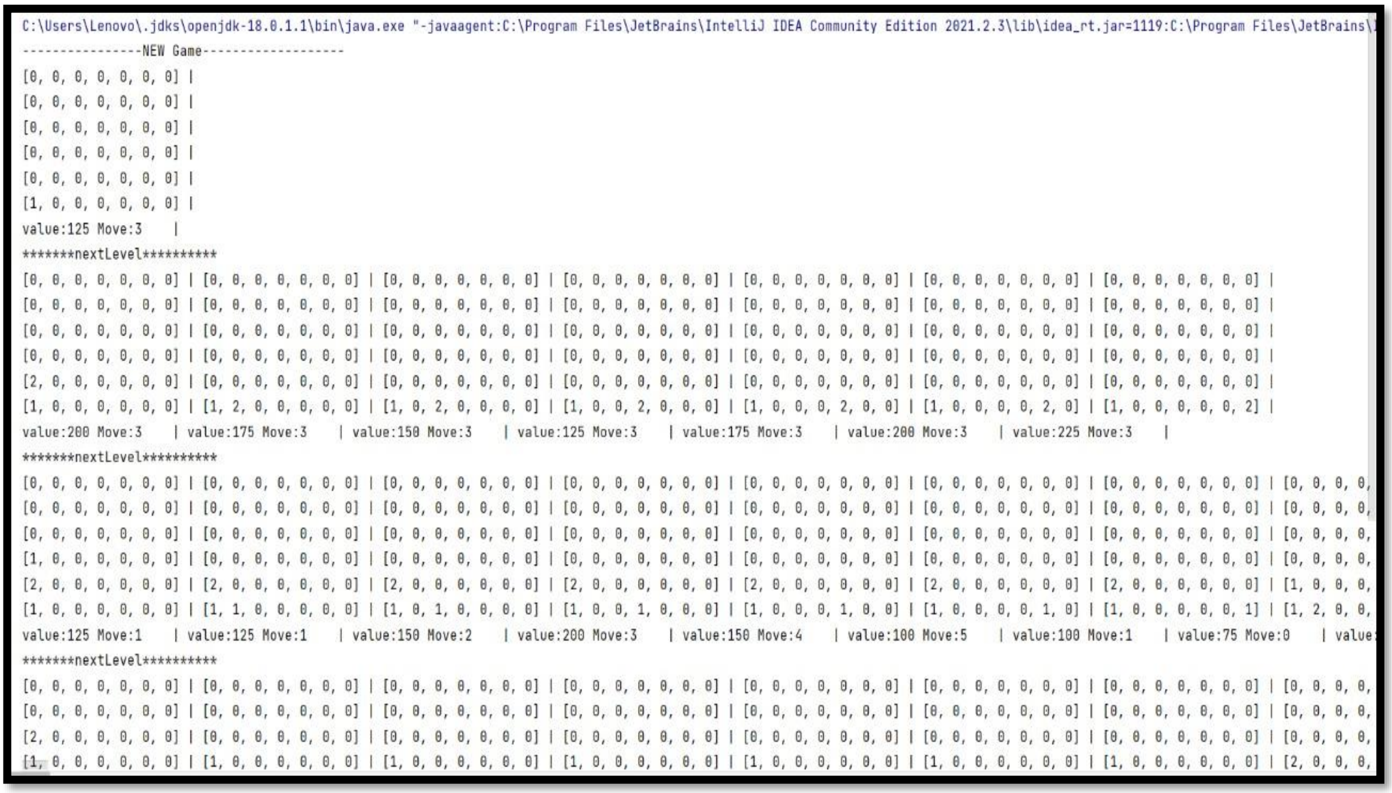
Gui Before :-



Gui After:

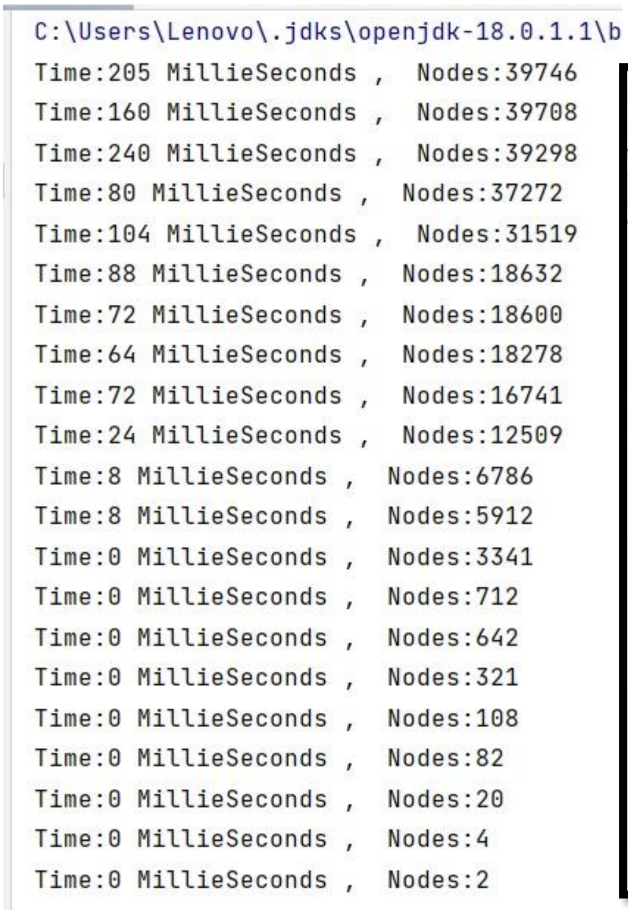


The output[tree]:-

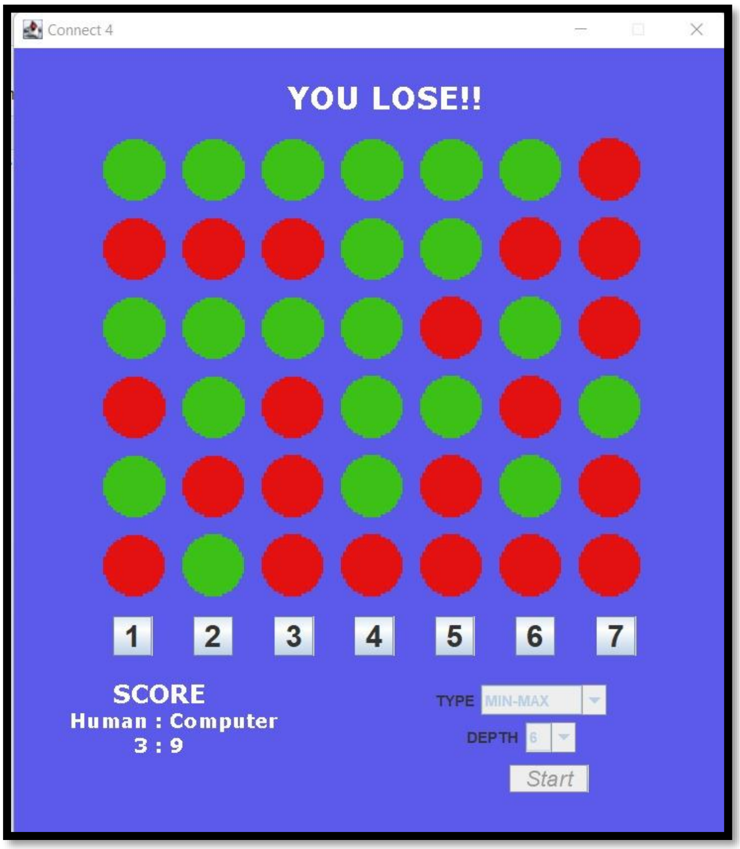


```
C:\Users\Lenovo\.jdks\openjdk-18.0.1.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.3\lib\idea_rt.jar=1119:C:\Program Files\JetBrains\I
----------------NEW Game-------------------
[0, 0, 0, 0, 0, 0, 0] |
[0, 0, 0, 0, 0, 0, 0] |
[0, 0, 0, 0, 0, 0, 0] |
[0, 0, 0, 0, 0, 0, 0] |
[0, 0, 0, 0, 0, 0, 0] |
[1, 0, 0, 0, 0, 0, 0] |
value:125 Move:3    |
*******nextLevel**********
[0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] |
[0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] |
[0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] |
[0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] |
[2, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] |
[1, 0, 0, 0, 0, 0, 0] | [1, 2, 0, 0, 0, 0, 0] | [1, 0, 2, 0, 0, 0, 0] | [1, 0, 0, 2, 0, 0, 0] | [1, 0, 0, 0, 2, 0, 0] | [1, 0, 0, 0, 0, 2, 0] | [1, 0, 0, 0, 0, 0, 2] |
value:200 Move:3    | value:175 Move:3    | value:150 Move:3    | value:125 Move:3    | value:175 Move:3    | value:200 Move:3    | value:225 Move:3    |
*******nextLevel**********
[0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0,
[0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0,
[0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0,
[1, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0,
[2, 0, 0, 0, 0, 0, 0] | [2, 0, 0, 0, 0, 0, 0] | [2, 0, 0, 0, 0, 0, 0] | [2, 0, 0, 0, 0, 0, 0] | [2, 0, 0, 0, 0, 0, 0] | [2, 0, 0, 0, 0, 0, 0] | [2, 0, 0, 0, 0, 0, 0] | [1, 0, 0, 0,
[1, 0, 0, 0, 0, 0, 0] | [1, 1, 0, 0, 0, 0, 0] | [1, 0, 1, 0, 0, 0, 0] | [1, 0, 0, 1, 0, 0, 0] | [1, 0, 0, 0, 1, 0, 0] | [1, 0, 0, 0, 0, 1, 0] | [1, 0, 0, 0, 0, 0, 1] | [1, 2, 0, 0,
value:125 Move:1    | value:125 Move:1    | value:150 Move:2    | value:200 Move:3    | value:150 Move:4    | value:100 Move:5    | value:100 Move:1    | value:75 Move:0    | value
*******nextLevel**********
[0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0,
[0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0,
[2, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0,
[1, 0, 0, 0, 0, 0, 0] | [1, 0, 0, 0, 0, 0, 0] | [1, 0, 0, 0, 0, 0, 0] | [1, 0, 0, 0, 0, 0, 0] | [1, 0, 0, 0, 0, 0, 0] | [1, 0, 0, 0, 0, 0, 0] | [1, 0, 0, 0, 0, 0, 0] | [2, 0, 0, 0,
```

The out put of The time and The Nodes expanded:-



```
C:\Users\Lenovo\.jdks\openjdk-18.0.1.1\b
Time:205 MillieSeconds ,  Nodes:39746
Time:160 MillieSeconds ,  Nodes:39708
Time:240 MillieSeconds ,  Nodes:39298
Time:80 MillieSeconds ,  Nodes:37272
Time:104 MillieSeconds ,  Nodes:31519
Time:88 MillieSeconds ,  Nodes:18632
Time:72 MillieSeconds ,  Nodes:18600
Time:64 MillieSeconds ,  Nodes:18278
Time:72 MillieSeconds ,  Nodes:16741
Time:24 MillieSeconds ,  Nodes:12509
Time:8 MillieSeconds ,  Nodes:6786
Time:8 MillieSeconds ,  Nodes:5912
Time:0 MillieSeconds ,  Nodes:3341
Time:0 MillieSeconds ,  Nodes:712
Time:0 MillieSeconds ,  Nodes:642
Time:0 MillieSeconds ,  Nodes:321
Time:0 MillieSeconds ,  Nodes:108
Time:0 MillieSeconds ,  Nodes:82
Time:0 MillieSeconds ,  Nodes:20
Time:0 MillieSeconds ,  Nodes:4
Time:0 MillieSeconds ,  Nodes:2
```



YOU LOSE!!

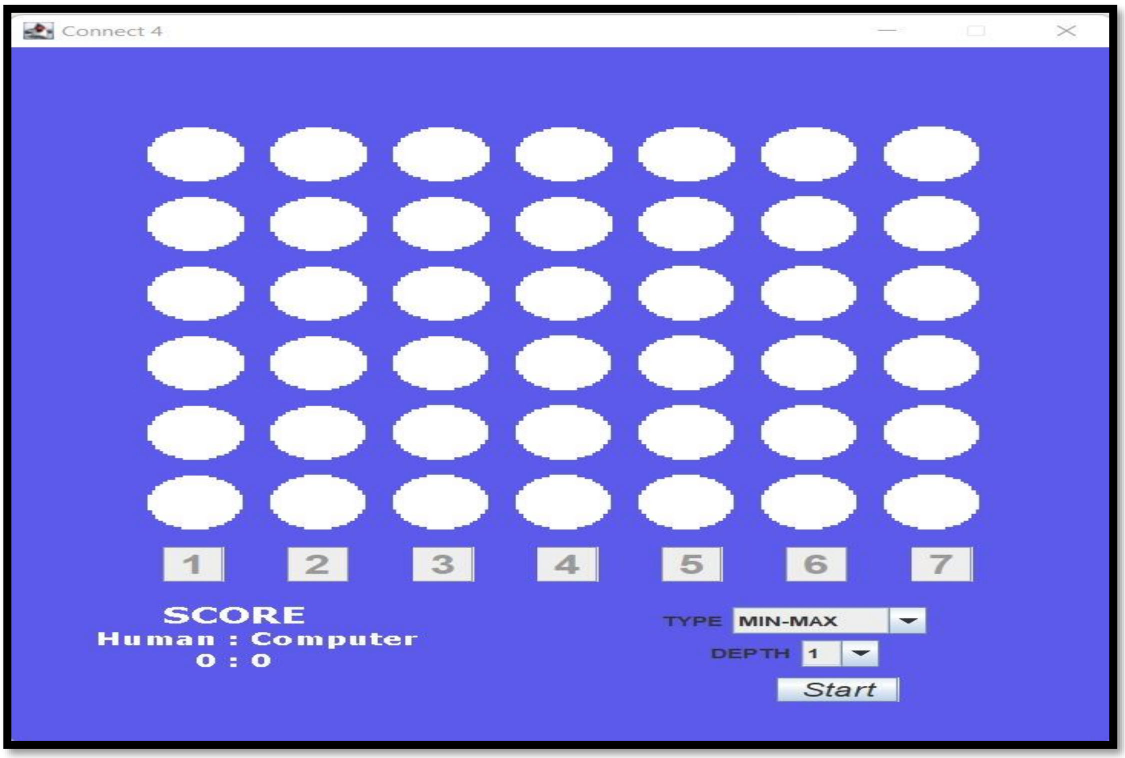
SCORE
Human : Computer
3 : 9

TYPE MIN-MAX
DEPTH 6

Minimax with alpha-beta pruning

Sample run Test

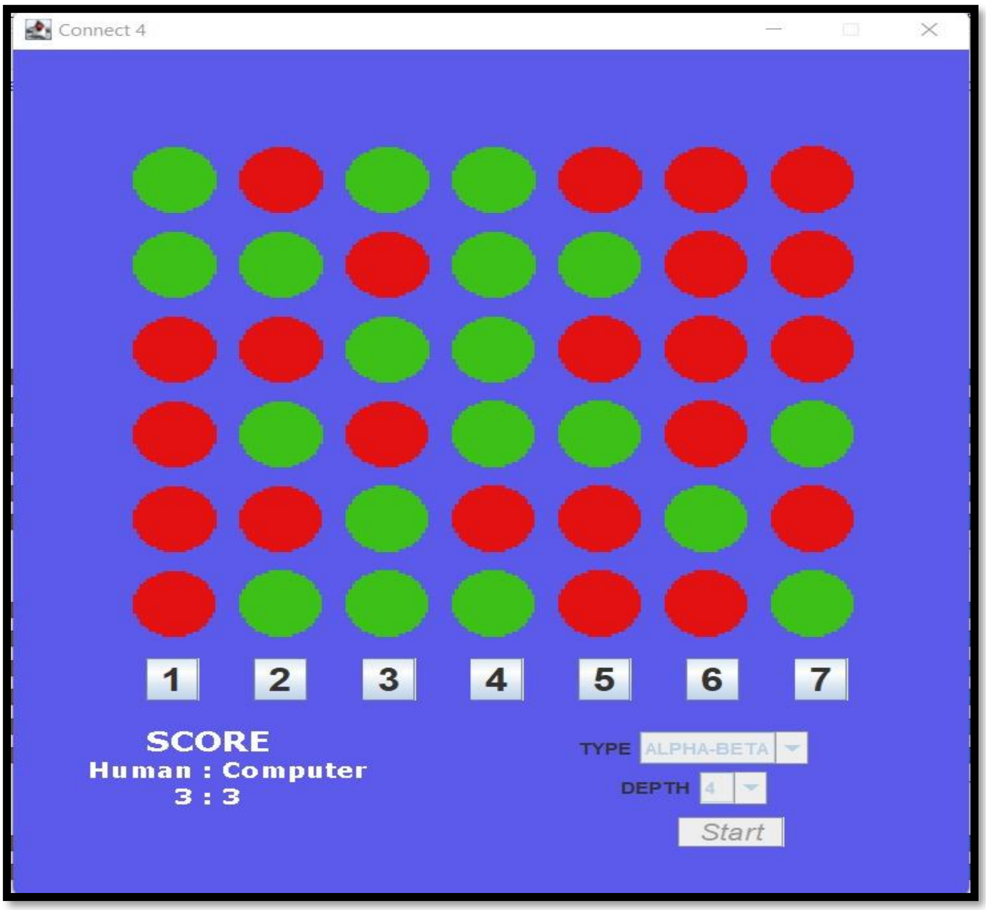
Gui Before :-



SCORE
Human : Computer
0 : 0

TYPE MIN-MAX
DEPTH 1

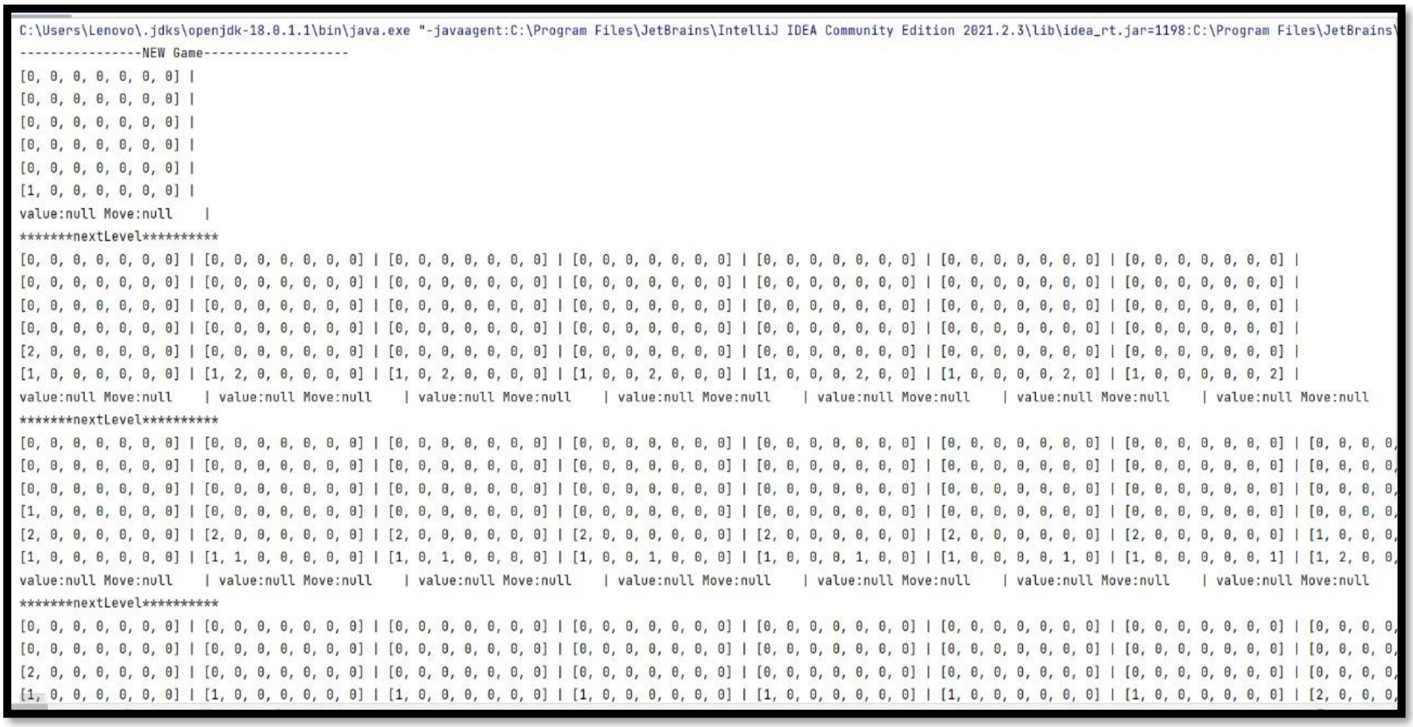
Gui After:



SCORE
Human : Computer
3 : 3

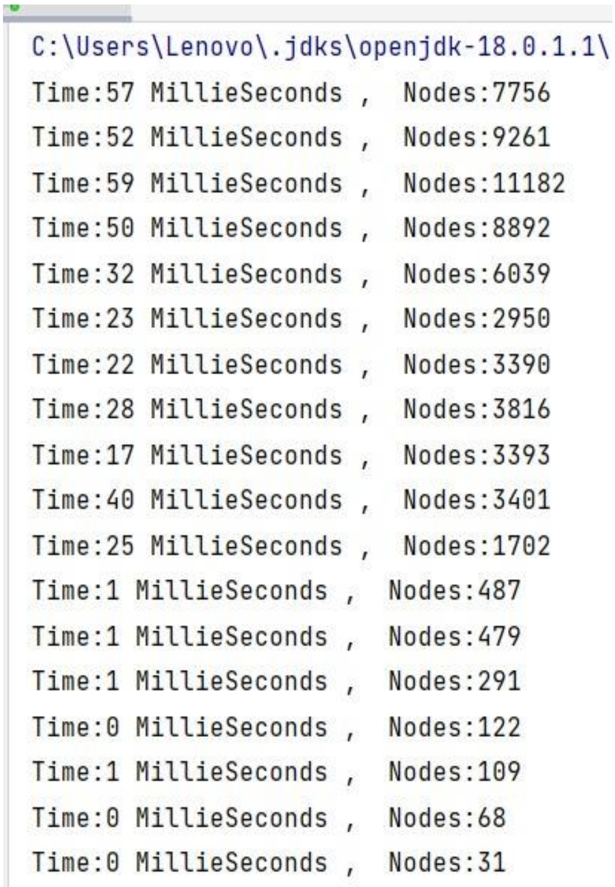
TYPE ALPHA-BETA
DEPTH 4

The output[tree]:-

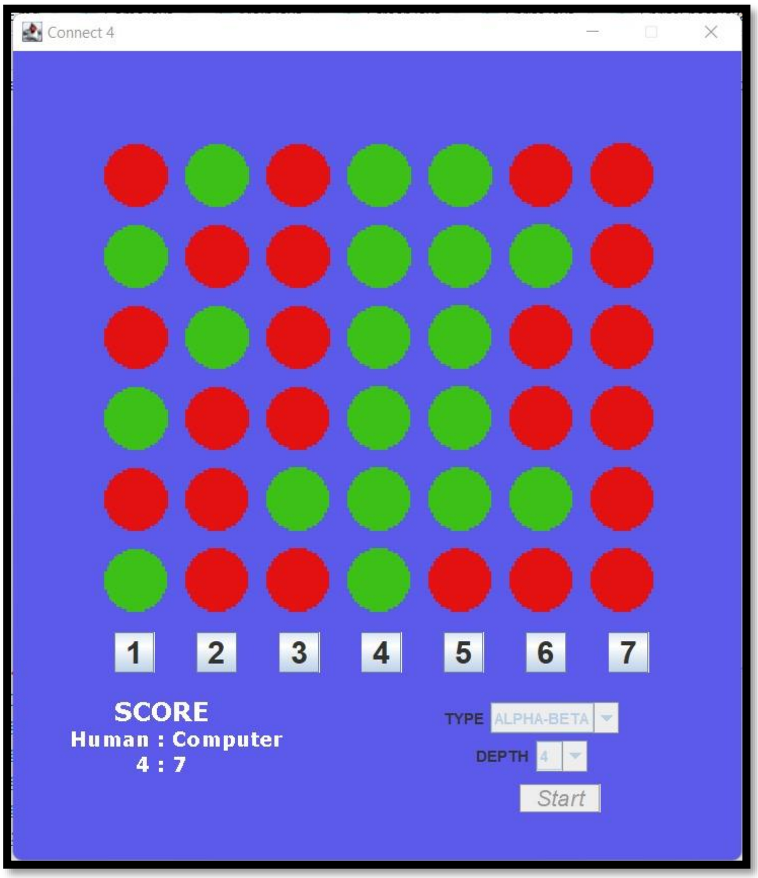


```
C:\Users\Lenovo\.jdks\openjdk-18.0.1.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.3\lib\idea_rt.jar=1198:C:\Program Files\JetBrains\
----------------NEW Game-------------------
[0, 0, 0, 0, 0, 0, 0] |
[0, 0, 0, 0, 0, 0, 0] |
[0, 0, 0, 0, 0, 0, 0] |
[0, 0, 0, 0, 0, 0, 0] |
[0, 0, 0, 0, 0, 0, 0] |
[1, 0, 0, 0, 0, 0, 0] |
value:null Move:null    |
*******nextLevel**********
[0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] |
[0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] |
[0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] |
[0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] |
[2, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] |
[1, 0, 0, 0, 0, 0, 0] | [1, 2, 0, 0, 0, 0, 0] | [1, 0, 2, 0, 0, 0, 0] | [1, 0, 0, 2, 0, 0, 0] | [1, 0, 0, 0, 2, 0, 0] | [1, 0, 0, 0, 0, 2, 0] | [1, 0, 0, 0, 0, 0, 2] |
value:null Move:null    | value:null Move:null    | value:null Move:null    | value:null Move:null    | value:null Move:null    | value:null Move:null    | value:null Move:null
*******nextLevel**********
[0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0,
[0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0,
[0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0,
[1, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0,
[2, 0, 0, 0, 0, 0, 0] | [2, 0, 0, 0, 0, 0, 0] | [2, 0, 0, 0, 0, 0, 0] | [2, 0, 0, 0, 0, 0, 0] | [2, 0, 0, 0, 0, 0, 0] | [2, 0, 0, 0, 0, 0, 0] | [2, 0, 0, 0, 0, 0, 0] | [1, 0, 0, 0,
[1, 0, 0, 0, 0, 0, 0] | [1, 1, 0, 0, 0, 0, 0] | [1, 0, 1, 0, 0, 0, 0] | [1, 0, 0, 1, 0, 0, 0] | [1, 0, 0, 0, 1, 0, 0] | [1, 0, 0, 0, 0, 1, 0] | [1, 0, 0, 0, 0, 0, 1] | [1, 2, 0, 0,
value:null Move:null    | value:null Move:null    | value:null Move:null    | value:null Move:null    | value:null Move:null    | value:null Move:null    | value:null Move:null
*******nextLevel**********
[0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0,
[0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0,
[2, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 0, 0] | [0, 0, 0, 0,
[1, 0, 0, 0, 0, 0, 0] | [1, 0, 0, 0, 0, 0, 0] | [1, 0, 0, 0, 0, 0, 0] | [1, 0, 0, 0, 0, 0, 0] | [1, 0, 0, 0, 0, 0, 0] | [1, 0, 0, 0, 0, 0, 0] | [1, 0, 0, 0, 0, 0, 0] | [2, 0, 0, 0,
```

The output of the time and nodes expanded:-



```
C:\Users\Lenovo\.jdks\openjdk-18.0.1.1\
Time:57 MillieSeconds ,  Nodes:7756
Time:52 MillieSeconds ,  Nodes:9261
Time:59 MillieSeconds ,  Nodes:11182
Time:50 MillieSeconds ,  Nodes:8892
Time:32 MillieSeconds ,  Nodes:6039
Time:23 MillieSeconds ,  Nodes:2950
Time:22 MillieSeconds ,  Nodes:3390
Time:28 MillieSeconds ,  Nodes:3816
Time:17 MillieSeconds ,  Nodes:3393
Time:40 MillieSeconds ,  Nodes:3401
Time:25 MillieSeconds ,  Nodes:1702
Time:1 MillieSeconds ,  Nodes:487
Time:1 MillieSeconds ,  Nodes:479
Time:1 MillieSeconds ,  Nodes:291
Time:0 MillieSeconds ,  Nodes:122
Time:1 MillieSeconds ,  Nodes:109
Time:0 MillieSeconds ,  Nodes:68
Time:0 MillieSeconds ,  Nodes:31
```

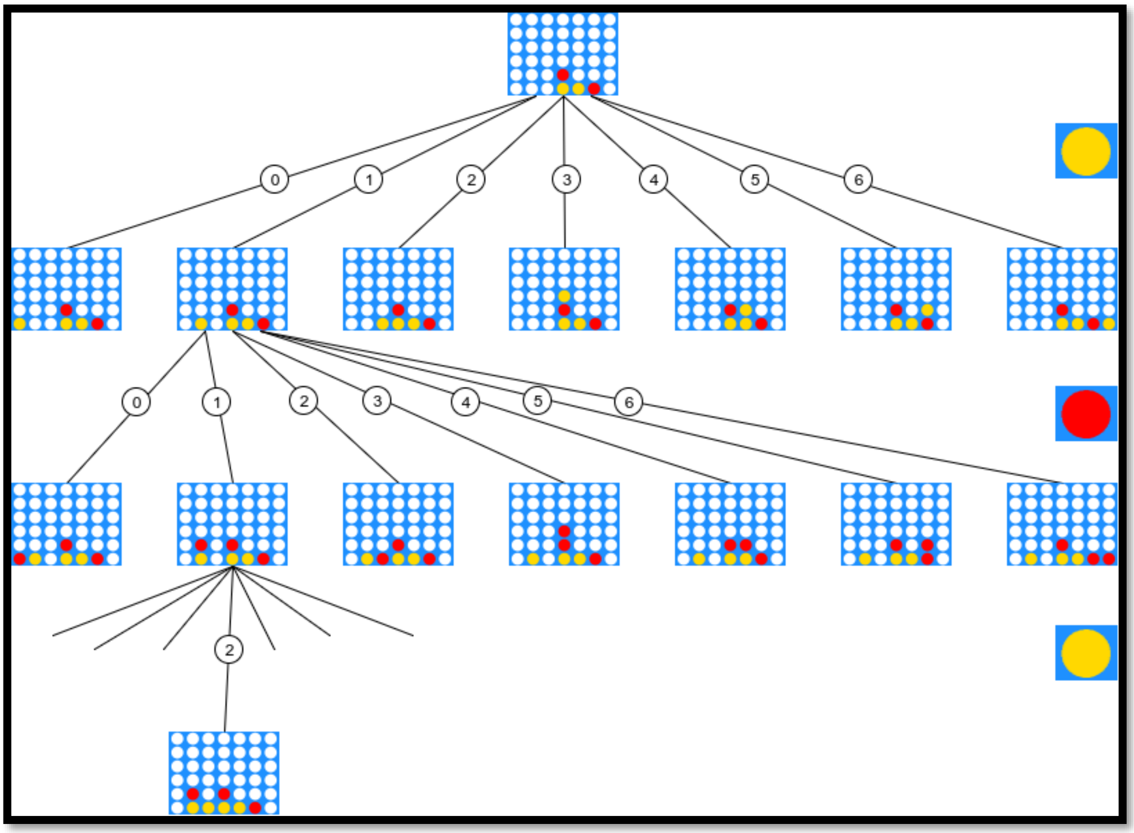


SCORE
Human : Computer
4 : 7

TYPE ALPHA-BETA
DEPTH 4

Comparison between time and nodes expanded in each mini-max:-

Time using alpha-beta pruning	Nodes Using alpha-beta pruning	Time Without alpha-beta pruning	Nodes Without alpha-beta pruning
57	7756	205	39746
52	9261	160	39708
59	11182	240	39298
50	8892	80	37272
32	6039	104	31519


0
1
2
3
4
5
6
0
1
2
3
4
5
6
2