CSE 537- ARTIFICIAL INTELLIGENCE

ASSIGNMENT -2

REPORT

CONNECTFOUR GAME IMPLEMENTATION

SUBMITTED BY:

SBU ID NAME

110338732 SADHANA KUMARAVEL

110349668 SREEPRADHANA THIRUMALAISWAMI

#NEWPLAYER IMPLIES USAGE OF BETTER EVALUATOR

OUTPUT FOR BASICPLAYER vs NEWPLAYER:

run_game(basic_player,new_player)

BASIC PLAYER:

NODES EXPANDED IN MINIMAX: 63758

Player 1 (☺) puts a token in column 0

0 1 2 3 4 5 6

0

1

2 ☺

3 ☻ ☻ ☻

4 ☺ ☺ ☻ ☺ ☻

5 ☺ ☻ ☺ ☺ ☺ ☻

NEW PLAYER:

NODES EXPANDED IN MINIMAX: 48822

Time: 0.393262

0

Player 2 (☻) puts a token in column 0

Total execution time: 18.741386

Win for ☻!

0 1 2 3 4 5 6

0

1

2 ☺

3 ☻ ☻ ☻ ☻

4 ☺ ☺ ☻ ☺ ☻

5 ☺ ☻ ☺ ☺ ☺ ☻

OUTPUT FOR NEWPLAYER vs ALPHABETA:
run_game(new_player,alphabeta_player)

NODES EXPANDED IN ALPHABETA 6618
Time: 0.103433
Player 2 (☻) puts a token in column 2

```
  0 1 2 3 4 5 6
0
1
2 ☺
3 ☻
4 ☻   ☻
5 ☻   ☺ ☺ ☺
```

NODES EXPANDED IN MINIMAX: 32356
Time: 0.535191
Player 1 (☺) puts a token in column 1
Total execution time: 5.192493
Win for ☺!

```
  0 1 2 3 4 5 6
0
1
2 ☺
3 ☻
4 ☻   ☻
5 ☻ ☺ ☺ ☺ ☺
```

OUTPUT FOR BASIC PLAYER vs ALPHABETA:
run_game(basic_player,alphabeta_player)
Time: 0.563744
NODES EXPANDED IN MINIMAX: 45020
Player 1 (☺) puts a token in column 1

```
  0 1 2 3 4 5 6
0
1
2 ☺   ☺ ☺ ☺
3 ☻   ☻ ☻ ☻
4 ☻ ☺ ☻ ☺ ☻   ☺
5 ☻ ☻ ☺ ☺ ☺ ☻ ☺
```

NODES EXPANDED IN ALPHABETA 7449
Time: 0.004626
Player 2 (☻) puts a token in column 1
Total execution time: 8.227341
Win for ☻!

0 1 2 3 4 5 6
0
1
2 ☺ ☺☺☺
3 ☻☻☻☻☻
4 ☻☺☻☺☻ ☺
5 ☻☻☺☺☺☻☺

MINIMAX:
Basic Evaluator vs Human Player:

NODES EXPANDED IN MINIMAX:100694
Time: 0.288727
Player 1 (☺) puts a token in column 0
Total execution time: 22.053239
Win for ☺!

0 1 2 3 4 5 6
0 ☺ ☺
1 ☺☺☺☺☻
2 ☻☻☺☺☺
3 ☺☺☻☻☻
4 ☻☻☺☺☻
5 ☻☻☺☺☻☻☻

New Player(Better Evaluator) vs Human Player:

NODES EXPANDED IN MINIMAX: 95806
Time: 1.569792
Player 1 (☺) puts a token in column 2
Total execution time: 23.619103
Win for ☺!

0 1 2 3 4 5 6
0 ☺
1 ☺ ☺
2 ☻ ☺ ☺ ☺ ☻
3 ☺ ☻ ☻ ☻ ☺
4 ☻ ☻ ☺ ☺ ☺ ☻ ☻
5 ☻ ☻ ☺ ☺ ☻ ☺ ☻

Process finished with exit code 0

Alpha beta vs Human player:
NODES EXPANDED IN ALPHABETA: 24517
Time: 0.094711
Player 1 (☺) puts a token in column 1
Total execution time: 4.816521
Win for ☺!

0 1 2 3 4 5 6
0
1 ☺
2 ☺ ☺ ☺ ☺
3 ☻ ☺ ☻ ☻
4 ☺ ☺ ☻ ☻ ☻
5 ☺ ☻ ☻ ☻ ☺
Process finished with exit code 0

Time indicates time of execution of every step
And Total execution time=> time from start to the completion of the game

Newplayer vs Human_player for k=5:

NODES EXPANDED IN MINIMAX: 59880
Time: 0.037324
0
Player 1 (☺) puts a token in column 0
Total execution time: 11.387319
Win for ☺!

 0 1 2 3 4 5 6
0
1 ☻
2 ☺☺☺☺☺☻
3 ☺☻☺☻☻☻
4 ☻☻☺☺☻☺
5 ☻☻☺☺☻☺

BETTER EVALUATOR FUNCTIONALITY:

In a ply, when the AI can choose between a child node to win and an other to block the opponent from winning, the basic evaluator chooses to block the opponent. However, the better evaluator is designed to find the column with longest chain along rows,columns and diagonals which may lead to victory and scores that column higher over the other for the AI to win the game at the earliest.
e.g.;

State after current move by AI player:
 0 1 2 3 4 5 6
0
1
2
3 ☺
4 ☺
5 ☻☺☺☻☻
Next ply: If a token is inserted by human player in column 5
Basic evaluator:
NODES EXPANDED IN MINIMAX: 44754
Time: 1.184268
Player 1 (☺) puts a token in column 6

 0 1 2 3 4 5 6
0
1
2
3 ☺
4 ☺
5 ☻☺☺☻☻☻☺

New Player(Better evaluator):
NODES EXPANDED IN MINIMAX: 38266
time: 1.19772
Player 1 (☺) puts a token in column 1
Total execution time: 5.914184
Win for ☺!

0 1 2 3 4 5 6
0
1
2 ☺
3 ☺
4 ☺
5 ☻☺☺☻☻☻
In the above example, the ply obtained using basic evaluator blocks the opponent by inserting token at column 6 rather than an intelligent move to win. However, the better evaluator reaches the goal state with a streak by choosing to insert token in column 1. Thus the better evaluator makes the AI much better than basic evaluator in terms of intelligence and streak to reach goal.

GENERALIZATION:
The game has been generalized for any k value. Specifying a third input in run_game under lab3.py takes the input as k value and executes the specified algorithm implementation for the same. If no input 'k' is specified in the run_game function call, the execution takes 4 as default k value.

ADDITIONAL WORK:

BOUND CHECKING:

The default setup doesn't check for the filling of column to its entirety. So when a column is scored the highest and it's already filled with either of the player values, an alternate column is to be chosen. Otherwise, there is a possibility for the program to end up in an infinite loop. Hence a check is made on the height of the column before considering it for the next move every time.