REPORT (20171189,20171134)

ALGORITHM:

- Alpha beta pruning is done for a limited depth d.
- At depth d , we stop searching by taking utility values based on heuristic function described below.
- We will choose the d value according to the time taken after implementing the algorithm.
- We use hash tables to store the states that we have already searched to avoid redundant paths.

Heuristic Calculation

- State S: Condition of both boards together after each player has his turn according to the rules mentioned.
- Utility (s) = w1 f1(s)+w2 f2(s)+ · · +wn fn (s)
- wi => weight of ith pattern
- fi(s) => number of times ith pattern occured in the state s.

WEIGHTS FOR EACH PATTERN

FOR MAX PLAYER(tries to maximise utility):

Wi	ith pattern
+100	three x in-a-row,column or diagonal
+10	two x in-a-row,column or diagonal (and empty cell)
+1	one x in-a-row,column or diagonal (two empty cells)
0	all other states

WEIGHTS FOR EACH PATTERN

FOR MIN PLAYER(tries to minimise utility):

Wi	ith pattern
-100	three o’s in-a-row,column or diagonal
-10	two o’s in-a-row,column or diagonal (and empty cell)
-1	one o’s in-a-row,column or diagonal (two empty cells)
0	all other states

Advantages

- This heuristic takes the advantage of how much is current state closer to the goal state of corresponding player.
- Using Hash tables avoids redundant paths in search tree.
- Using alpha beta pruning reduces the branches to be searched.
- Searching upto a depth d helps in retreiving the best possible move in given time limit and given depth.