

Algorithm:

def score (game)

if game.win? @ player)

seturn 10

else if game.win? @ opponent)

Return - (0

return 0

end

```
det minar (game)
       return riore (game) if Igame ovet?
      mores = { ]
       moves = []
             offen beto ( self books, col, who boto venery of
 function minimax (board, depth, istaximizing):
      if game_over (board):
          setuen evaluate (board)
                                          Maring . Ligarian )
                                  (w-) trail - low . won
      : f is Maximizins :
         best score = - 00
         for each empty cell (raw, col) in board:
              simulate more (board, sow, cal, x')
              score = minimax (board depth +1, False)
 word) duas . mida move (board, row, col)
             best - score = man (best noce, wore)
       setuen best scole a from the sond
    elle:
        best_wore = +00 and land a los won
        for each empty cell (sow, col) is board:
            simulate_move (board, saw, col, 'o')
            Rose = minimox (board, lepth +1 True)
            undo- more (board, sow, col)
            best score = min (best score, more)
                              server non not but being t
       seturn best score
function find best more (board, is Mosimuzing):
    best_move = (-1,-1)
    best-score = -00 if is Maximizing else 700
    for each empty cell (sou, cul) in board:
         Rinulate_nove (board, now, cot, it if is Maximizing, else '0')
         move_score = minimax (boaled, 0, not is Maximing)
        undo_move (board, gaw, col)
         if is Maximizing and move more > best since:
                                                           highwo
               best_scole = more_ uole
               best_more = (sourcol)
        elif not is Morninging and move scale < beet wars
              best score = move - more
              best-more = ( sow, col)
  seturn best more
```