Practical-2

Write a program to implement Tic-Tac-Toe game for two players, X(odd turn) and O(even) who take turn making the spaces in 3x3 grid: The player who succeeds in placing three respective marks in horizontal or vertical or diagonal wins the game .

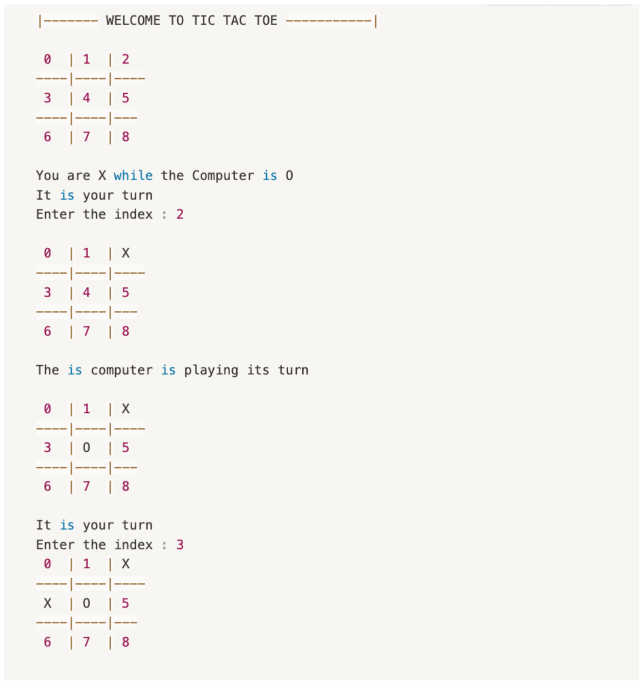
[1]:
```
# 1. Display the  3x3 Tic-Tac-Toe board
def print_board(s,index=0):
    if s == BOARD_PLAYER_X:
      return 'X'
    if s == BOARD_PLAYER_O:
      return 'O'
    return str(index)

def  display_board(board):
    print(" " + print_board(board[0],0) + " | " + print_board(board[1],1) + " |␣
 ↪" + print_board(board[2],2) + "  ")
    print("---|---|---")
    print(" " + print_board(board[3],3) + " | " + print_board(board[4],4) + " |␣
 ↪" + print_board(board[5],5) + "  ")
    print("---|---|---")
    print(" " + print_board(board[6],6) + " | " + print_board(board[7],7) + " |␣
 ↪" + print_board(board[8],8) + "  ")
```

[2]:
```
# 2. Initalizing the Params
BOARD_EMPTY = 0
BOARD_PLAYER_X = 1
BOARD_PLAYER_O = -1

import time
state = [0, 0, 0, 0, 0, 0, 0, 0, 0]
```

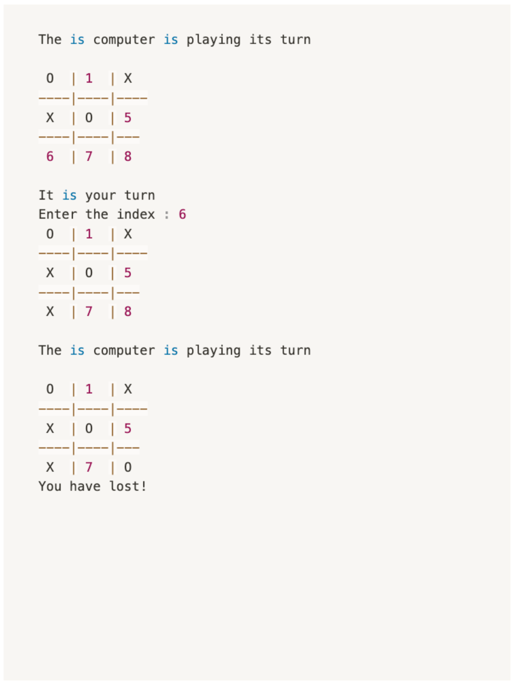
[3]:
```
# 3. Initializing the State and Player
from collections import Counter
def player(s):
  counter = Counter(s)
  x_places = counter[1]
  o_places = counter[-1]
  if x_places + o_places == 9:
    return None
  elif x_places > o_places:
    return BOARD_PLAYER_O
  else:
    return BOARD_PLAYER_X

def actions(s):
  play = player(s)
  actions_list = [(play, i) for i in range(len(s)) if s[i] == BOARD_EMPTY]
  return actions_list

def result(s, a):
  (play, index) = a
  s_copy = s.copy()
  s_copy[index] = play
  return s_copy

def terminal(s):
  for i in range(3):
    # Checking if a row is filled and equal.
    if s[3 * i] == s[3 * i + 1] == s[3 * i + 2] != BOARD_EMPTY:
      return s[3 * i]
    # Checking if a column is filled and equal.
    if s[i] == s[i + 3] == s[i + 6] != BOARD_EMPTY:
      return s[i]
  if s[0] == s[4] == s[8] != BOARD_EMPTY:
    return s[0]
  if s[2] == s[4] == s[6] != BOARD_EMPTY:
    return s[2]
  # Checking if the game has no more moves available
  if player(s) is None:
    return 0
  # Return None if none of the previous conditions satisfy.
  return None
```

[4]:
```
# 4. Minmax Algorithm and utility function
def utility(s, cost):
  term = terminal(s)
  if term is not None:
    return (term, cost)
  action_list = actions(s)
  utils = []
  for action in action_list:
    new_s = result(s, action)
    utils.append(utility(new_s, cost + 1))

  score = utils[0][0]
  idx_cost = utils[0][1]
  play = player(s)
  if play == BOARD_PLAYER_X:
    for i in range(len(utils)):
     if utils[i][0] > score:
       score = utils[i][0]
       idx_cost = utils[i][1]
  else:
    for i in range(len(utils)):
      if utils[i][0] < score:
        score = utils[i][0]
        idx_cost = utils[i][1]
  return (score, idx_cost)

def minimax(s):
  action_list = actions(s)
  utils = []
  for action in action_list:
    new_s = result(s, action)
    utils.append((action, utility(new_s, 1)))
  if len(utils) == 0:
    return ((0, 0), (0, 0))
  sorted_list = sorted(utils, key=lambda l : l[0][1])
  action = min(sorted_list, key = lambda l : l[1])
  return action
```

[5]:
```
# 5. Driver Function
__name__ == '__main__'
if __name__ == '__main__':
  # Initializing the state
  s = [ BOARD_EMPTY for _ in range(9)]
  print('|------- WELCOME TO TIC TAC TOE -----------|')
  print()
  display_board(s)
  print()
  print('You are X while the Computer is O')
  while terminal(s) is None:
    play = player(s)
    if play == BOARD_PLAYER_X:
      # Take input from user
      print('\nIt is your turn')
      index = int(input(f"Enter the index : "))
      if not s[index] == BOARD_EMPTY:
        print('That coordinate is already taken. Please try again.')
        continue
      s = result(s, (BOARD_PLAYER_X, index))
      display_board(s)
      time.sleep(1)
    else:
      print('\n\nThe is computer is playing its turn')
      action = minimax(s)
      s = result(s, action[0])
      display_board(s)
  winner = terminal(s)
  if winner == BOARD_PLAYER_X:
    print("You have won!")
  elif winner == BOARD_PLAYER_O:
    print("You have lost!")
  else:
    print("It's a tie.")
```



```
|------- WELCOME TO TIC TAC TOE -----------|

 0  | 1  | 2
----|----|----
 3  | 4  | 5
----|----|---
 6  | 7  | 8

You are X while the Computer is O
It is your turn
Enter the index : 2

 0  | 1  | X
----|----|----
 3  | 4  | 5
----|----|---
 6  | 7  | 8

The is computer is playing its turn

 0  | 1  | X
----|----|----
 3  | O  | 5
----|----|---
 6  | 7  | 8

It is your turn
Enter the index : 3
 0  | 1  | X
----|----|----
 X  | O  | 5
----|----|---
 6  | 7  | 8
```



```
The is computer is playing its turn

 O  | 1  | X
----|----|----
 X  | O  | 5
----|----|---
 6  | 7  | 8

It is your turn
Enter the index : 6
 O  | 1  | X
----|----|----
 X  | O  | 5
----|----|---
 X  | 7  | 8

The is computer is playing its turn

 O  | 1  | X
----|----|----
 X  | O  | 5
----|----|---
 X  | 7  | O
You have lost!
```