Lab Topic 11 - Designing an AI Agent for the Tic-Tac-Toe Game

CNM Peralta

Background

Tic-Tac-Toe is a paper-and-pencil game for two players, X and O, who take turns marking spaces in a 3 × 3 grid. The first player to place three eight-connected markers wins the game (either 3 X's for Player X or 3 O's for Player O). Tic-Tac-Toe has 26,830 possible games. Despite this vast variety of possible games, and assuming that Player X plays first, there are 91 unique end-games where Player X wins, 44 unique end-games where Player O wins, and 3 unique end-games that are draws.

Tic-Tac-Toe is one of the games that is commonly used to demonstrate the usage of the minimax algorithm and/or alpha-beta pruning because its state space is still reasonably within the limits of modern day consumer computers.

The pseudocode for minmax is:

```
value(s)
  if s is terminal: return utility(s)
  if s is max_node: return max_value(s)
  if s is min_node: return min_value(s)
```

```
max_value(s)
  m = neg_inf
  //action, a, s’ = result(s, a)
  for a, s’ in successors(s) v =
  value(s’)
    m = max(m, v)
  return m
```

```
min_value(s)
  m = pos_inf
  //action, a, s’ = result(s, a)
  for a, s’ in successors(s) v =
  value(s’)
    m = min(m, v)
  return m
```

Exercise

The task is to create an AI player that will play against a human player. To get full points, your AI should never lose; it will always win or get a draw. The AI should be able to give a/an [intelligent] move given the human player's move. A UI is required.

The breakdown of points for the exercise is as follows:

Criterion	**Points**
Playable user interface	5
Use of minimax algorithm	10
Total Score	15