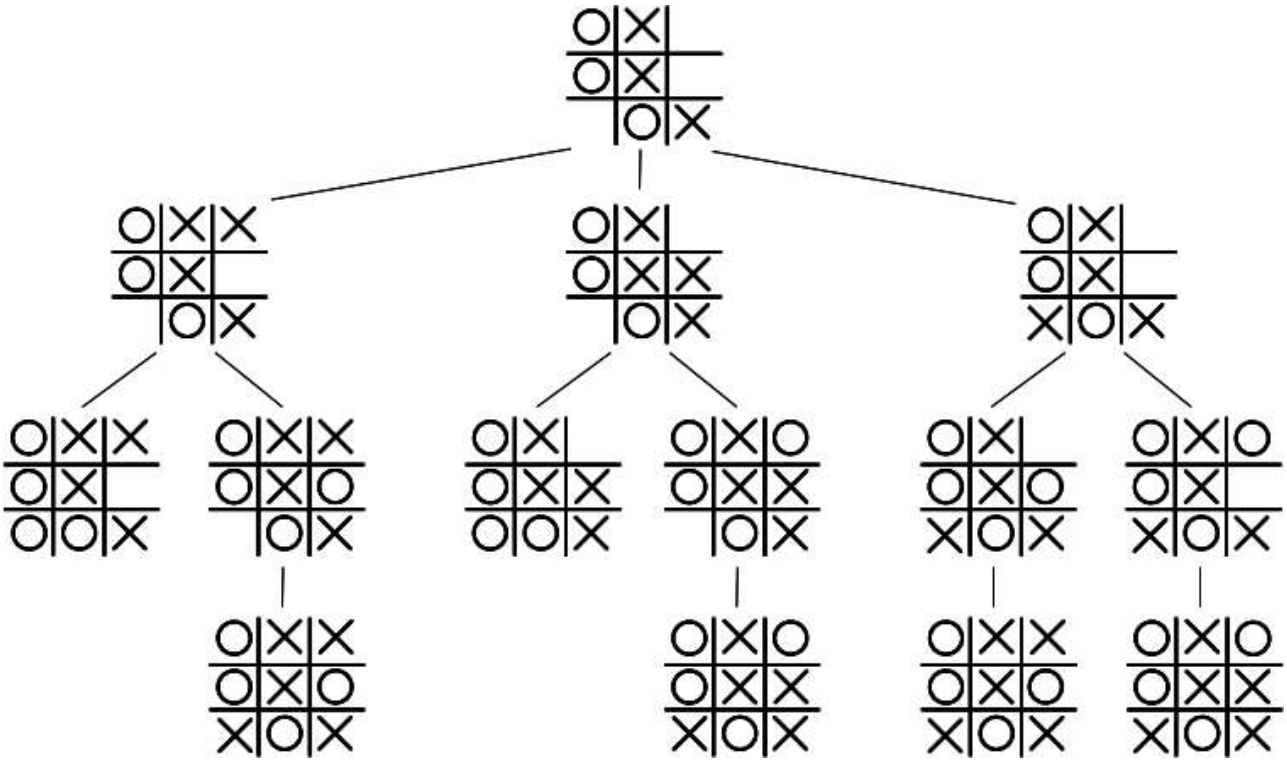
Minimax

Help Center

Tic-Tac-Toe Game Tree

The following image shows a portion of the game tree for Tic-Tac-Toe:



In the tree, each node represents a board position, and the children of each node represent possible moves from that board position. Therefore, this tree illustrates all possible ways that a game could be played out if you start from the board position at the root node.

Minimax

Let's consider using Minimax on the game tree above. We will assign each board a score as follows: if player X wins, the board is scored +1, if player O wins, the board is scored as -1, and if the game is a draw, the board is scored as 0. Note that with this simple scoring function, you can only directly assign scores to board positions that represent completed games, which are the six bottom leaf nodes in the tree.

A key idea behind Minimax is that you are assuming that both players will make the best move available to them. We assume that they will minimize the maximum loss that their opponent can cause. One player is always trying to create a score of -1 (the minimum possible score), while the other player is always trying to create a score of +1 (the maximize possible score). Given the scoring function above, player X benefits from positive scores, so is the maximizing player, and player O benefits from negative scores, so is the minimizing player.

Given the board position at the root node, it is player X's turn, so we should pick the move that results in the maximum score. As the tree shows, player X has three possible moves. In order to decide which

move to select, we would call minimax on the board that results from each move. That board would then become the root of the game tree and the process would repeat itself. Except at this level, it is player O's turn, so we would pick the move that results in the minimum score at this level. If you consider the left-most child of the root node (in which X plays in the upper right hand corner), player O has two possible moves. In order to decide which move O should select, we would again call minimax on the board that results from each move. This process repeats recursively until you get to a board in which the game is over. You can then use the scoring function to determine the score of the board directly, without needing to minimize or maximize anything.

Note that if you had a more sophisticated scoring function, you could potentially score any board. This is helpful for games that have very large game trees that would make it impossible to search the entire tree. In that case, you would generally stop searching at a specified depth and score all of the boards at that depth. You then use the same minimizing / maximizing process to determine which moves to select from there.

Created Sun 13 Jul 2014 2:44 PM PDT
Last Modified Wed 30 Jul 2014 11:43 AM PDT