**MINIMAX**

Evaluating Leaves

We now know that we can evaluate the leaves of a game tree, but how does that help us? How are we going to use those values to find the best possible move for a game state that isn’t a leaf?

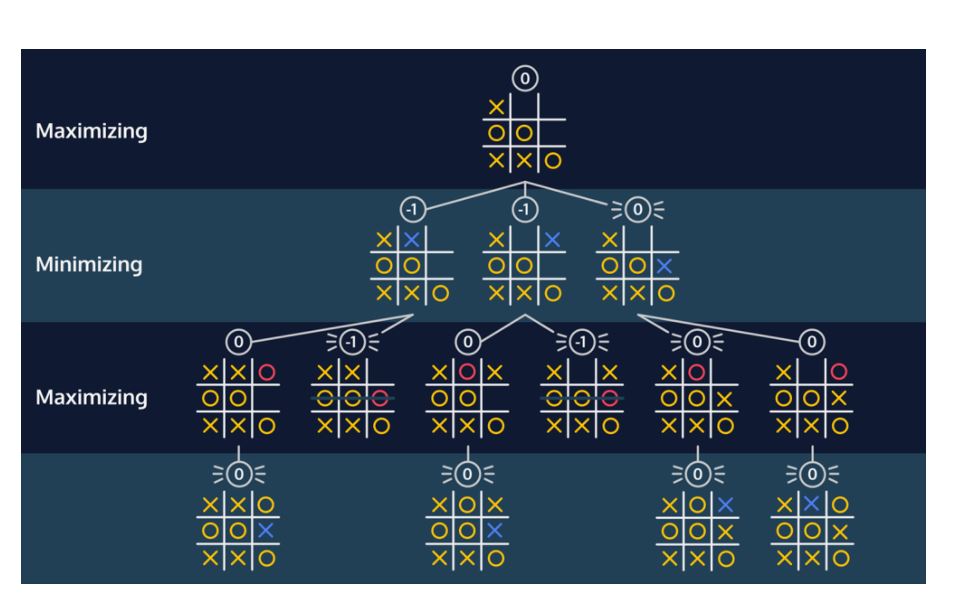
Let’s imagine a situation where you’re playing as the "X" player in a game of Tic-Tac-Toe and the game is almost over. The game board isn’t a leaf but it’s close. You have three possible moves. All three moves will immediately end the game — each of those future boards will be leaves.

Let’s say picking move *A* will result in you winning and moves *B* and *C* will each result in a tie. You’d clearly pick move *A*.

By picking move *A*, you’ve picked the move that led to the board with the *highest* value. You were picking between a 1 (an "X" win) or two 0s (the moves that would lead to a tie). Because you picked the move with the highest value, we can say that "X" is the *maximizing player*.

Let’s say you were playing a the "O" player under the same circumstances. Picking move *A* would somehow immediately lead to "X" winning, while moves *B* and *C* would lead to a tie. You’d pick one of the boards that would lead to a tie. "O" is the *minimizing player*. You would love to pick a board with the value of -1 (an "O" win), but unfortunately, that board doesn’t exist. You’ll have to settle with picking a board with the value of 0. At least you prevent "X" from winning.

Take a look at the gif below to see the minimax algorithm running on a game of Tic-Tac-Toe where the tree is slightly bigger. The leaf nodes get evaluated and those values get passed up the tree depending based on whether it was the minimizing or maximizing player’s turn.



The root of the tree ends up with a value of 0. This means that the best move for the current player is the move that corresponded with the value of 0.

### Instructions

We’ve made an applet that will walk you through the minimax algorithm.

At the top of the tree, you’ll see the current game state. You’re the maximizing player ("X") trying to figure out the optimal move.

Begin by evaluating the leaf nodes of the tree. If you evaluate a node correctly, the drop down box will turn green. You can evaluate non-leaf nodes once every child of that node has been correctly evaluated.

The value of a non-leaf will depend on whether it is the maximizing player or minimizing player’s turn.

Bubble the values to the top of the tree to find the value of the optimal move.

