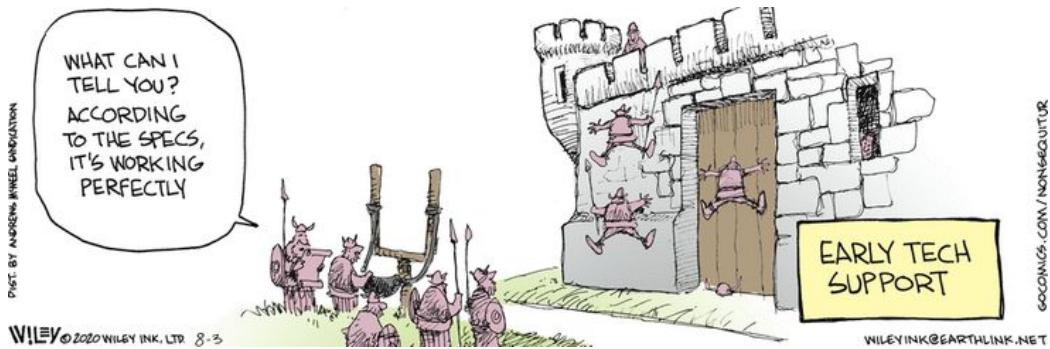


**CSCI 3202**  
**Lecture 21**  
**October 15, 2025**



Non Sequitur by Wiley Miller: <https://www.gocomics.com/nonsequitur>

### Announcements

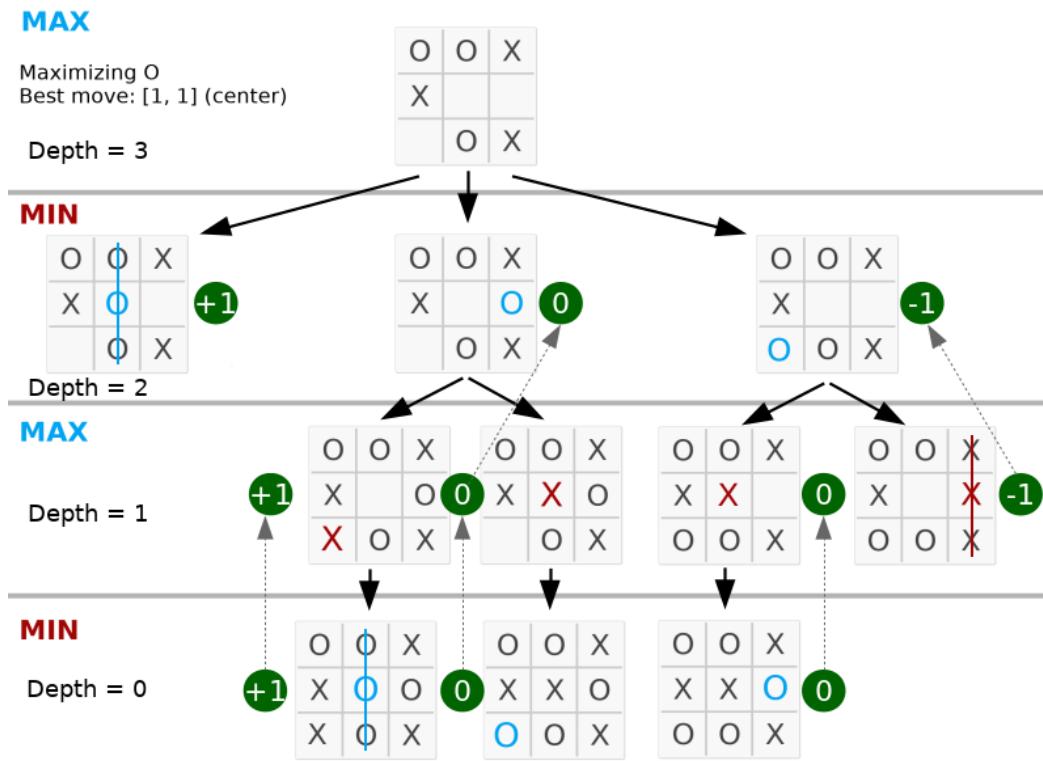
- Homework 6 will be released on Friday
  - Due in 1 Week on Friday, 10/24 by 11:59 pm
- Quiz 7 on Friday
  - Game Trees
  - Minimax
  - Alpha Beta pruning

### An Interesting Evaluation Function

- For a game with a limited tree size we can use the following evaluation function **only at terminal nodes**
- If a node is not terminal, we don't evaluate it

$$f(x) = \begin{cases} 1 & \text{if terminal node is a win} \\ 0 & \text{if terminal node is a draw} \\ -1 & \text{if terminal node is a loss} \end{cases}$$

- We end up with a lot of ties



### Alpha Beta pruning

- Minimax is a backtracking algorithm therefore we have to go through the entire tree
- The size of the tree grows exponentially with each layer
- There are some nodes or entire branches of the tree that will never be the result of minimax. We can stop evaluating these nodes or branches to save time
- Alpha Beta is recursive. It follows a similar evaluation path to DFS search
- Remember that the algorithm descends to the bottom of the tree then works back up. This means that alpha beta is doing its primary work as it climbs back up the tree
  - You don't get the utility function values until you get to the bottom of the tree
- Each time the a player moves, we reevaluate the tree with either minimax or alpha beta
- [Detailed Alpha Beta Pruning.pdf](#)

### Questions

- How do the result of Minimax differ from the results of Alpha Beta pruning?
- Why does the order of evaluation affect Alpha Beta pruning?

- How can we determine the sequence of best moves from a particular position?
- What happens if your opponent makes a non-optimal move?

### **Upcoming**

- Alpha-Beta pruning examples on Friday
- Project description on Friday
  - Default project is Mancala
  - Can create your own project as long as it uses one of the topics in our class
  - Can work in teams of 2 if desired