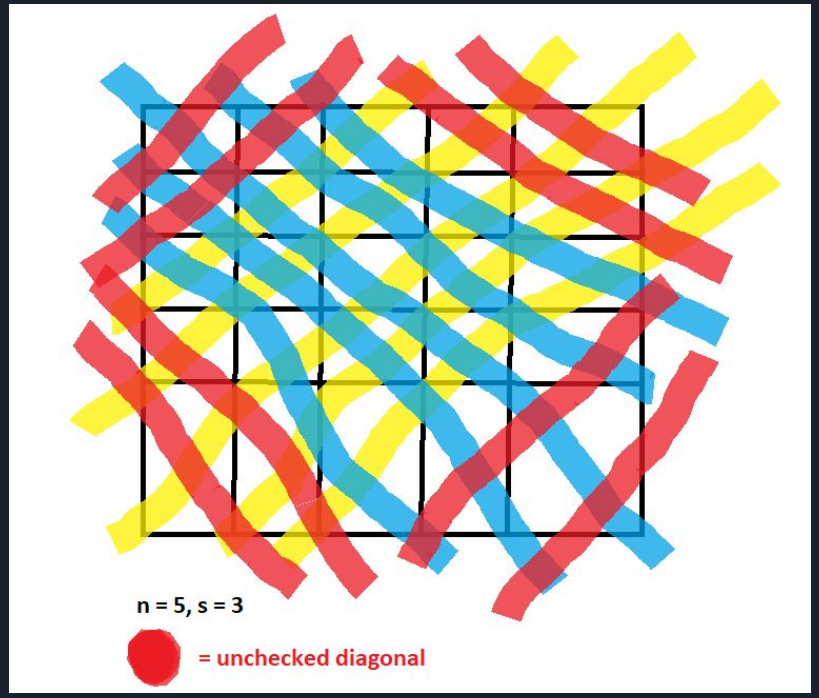
Minimax and Alphabeta

By: Jason Botzas-Coluni, Patrick Godbout and William Mongeau

Heuristic 1

- For every row and column in the board we iterate over each spot in the game board.
- If the spot contains an X, we increase the score of the node by 1, if it contains an O we decrease the score of the node by 1
- If the spot we previously visited was of the same type as the current spot, we increase by 1 + score of previous spot
- For example: a row containing [O, ., X, X, X]
    - O -> -1
    - . -> 0
    - X -> 1
    - X -> 1 + 1 = 2
    - X -> 1 + 2 = 3
    - Total score of this row: 5
- For diagonals, we follow the same idea, but with only diagonals that can possible we won on. That is, any diagonal on the board with a length >= s
- Once all rows, columns, and diagonals are evaluated we sum their scores together to return a score for the whole board

Heuristic 1



Heuristic 2

- Heuristic 2 generates every single sub-row, sub-column and sub-diagonal of size s, and then increases the heuristic for every sub-row, sub-column and sub-diagonal that contains only X's by the amount of X's in that sub-row, sub-column or sub-diagonal
- Similarly, it decreases the heuristic for every single sub-row, sub-column and sub-diagonal of size s that only contains O's by the amount of O's in that sub-row, sub-column or sub-diagonal
- For example:
  - [X, ., X, X] -> 3
  - [X, ., O, X] -> 0
  - [X, -, X, X] -> 0
  - [O, ., O, .] -> -2
- Similarly to heuristic 1, this evaluation function only checks diagonals that can be won on, that is diagonals whose lengths are >= s

Heuristic 1 vs Heuristic 2

- The size n of the board is 8, the number of blocks b is 6, the number of connected pieces s is 5 and the maximum time for evaluation t is 5
  - The average time for heuristic 1 was: 0.0003
  - The average time for heuristic 2 was: 0.0004
- The size n of the board is 5, the number of blocks b is 4, the number of connected pieces s is 4 and the maximum time for evaluation t is 5
  - The average time for heuristic 1 was: 9.140050516647492e-05
  - The average time for heuristic 2 was: 9.336470104290115e-05
- Although both heuristic functions evaluate at approximately the same speeds, heuristic 1 tends to outperform heuristic 2. This is because it favours game states that connect as many game pieces as possible and does not need as much future state information to optimize its move

Effects of different depths

- While shallower depths improve the run time of the games, they harm the decision making of the AIs
- This is because the AIs are not looking as far into the “future” with shallower depths
- As their decisions are less informed there is no guarantee that the decision they make will lead to the optimal decision to be made in the future
- For example: The choice the AI makes promises a win when checking at depth 3, however when checking depth 4 the only possible move that it’s opponent would make would block the AI’s initial choice. The decision that the AI made only looking at depth 3 would have been completely ignored if depth 4 was used instead.

Minimax vs Alphabeta

- Alphabeta reduces dramatically the amount of nodes that are visited
- For a board of size 4, minimax evaluates heuristics on 4 441 523 nodes
- For a board of size 4, alphabeta evaluates heuristics on 59 983 nodes
- This difference results in shorter games using alphabeta on equivalent boards.
- Furthermore, alphabeta doesn’t affect performance negatively, as it only prunes branches it know will not help it win