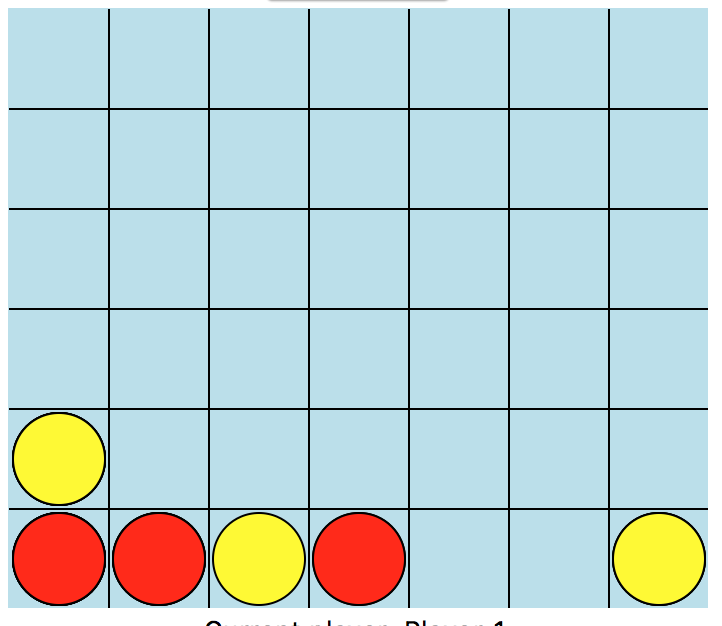
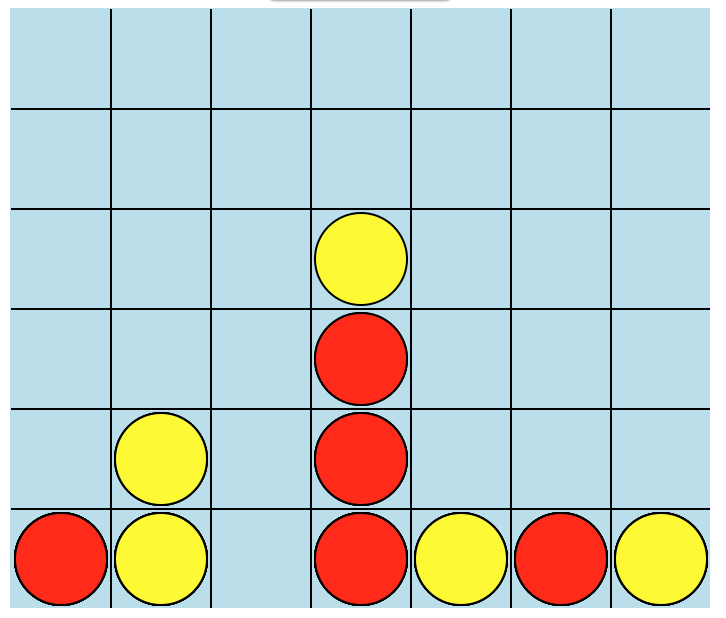
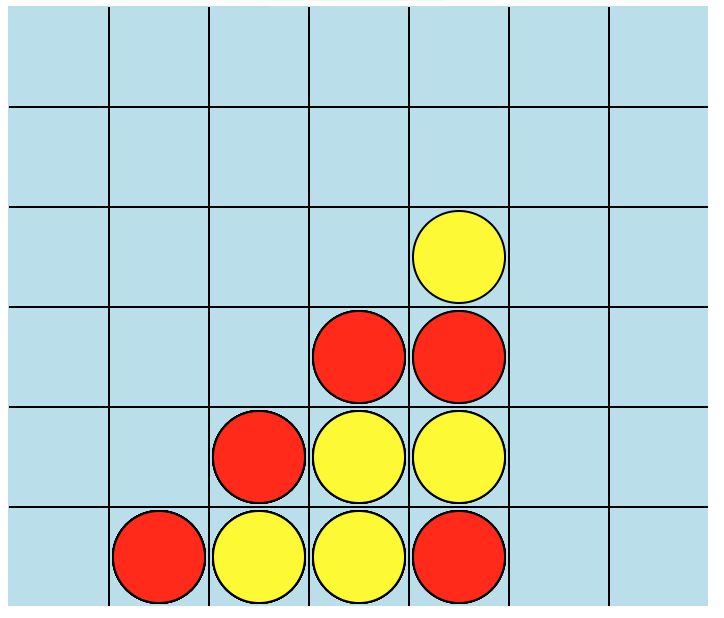
Examples of conditions under which the alpha beta search finds the opportunity for red to win, and makes the correct move to block this win







**What We Used**

**Minimax** is a decision rule used in [decision theory](https://en.wikipedia.org/wiki/Decision_theory), [game theory](https://en.wikipedia.org/wiki/Game_theory), [statistics](https://en.wikipedia.org/wiki/Statistics) and [philosophy](https://en.wikipedia.org/wiki/Philosophy) for *mini*mizing the possible [loss](https://en.wikipedia.org/wiki/Loss_function) for a worst case (*max*imum loss) scenario. Originally formulated for two-player [zero-sum](https://en.wikipedia.org/wiki/Zero-sum) [game theory](https://en.wikipedia.org/wiki/Game_theory), covering both the cases where players take alternate moves and those where they make simultaneous moves, it has also been extended to more complex games and to general decision-making in the presence of uncertainty (Norvig).

What would help

**Alpha–beta pruning** is a [search algorithm](https://en.wikipedia.org/wiki/Search_algorithm) that seeks to decrease the number of nodes that are evaluated by the [minimax algorithm](https://en.wikipedia.org/wiki/Minimax#Minimax_algorithm_with_alternate_moves) in its [search tree](https://en.wikipedia.org/wiki/Game_tree). It is an [adversarial search algorithm](https://en.wikipedia.org/w/index.php?title=Adversarial_search_algorithm&action=edit&redlink=1) used commonly for machine playing of two-player games ([Tic-tac-toe](https://en.wikipedia.org/wiki/Tic-tac-toe), [Chess](https://en.wikipedia.org/wiki/Chess), [Go](https://en.wikipedia.org/wiki/Go_(board_game)), etc.). It stops completely evaluating a move when at least one possibility has been found that proves the move to be worse than a previously examined move. Such moves need not be evaluated further. When applied to a standard minimax tree, it returns the same move as minimax would, but prunes away branches that cannot possibly influence the final decision.

**Our Next Steps**

Implement a heuristic

1. Computational cost. Since every move requires running the heuristic thousands of times if it is too complicated, the depth of searches will be adversely affected.
2. Correspondence to wins. If a heuristic doesn't provide useful about the game, there isn't any reason to use one.
3. Ease of Implementation. More complicated heuristics take more time to program so they run quickly and without errors (Pearce) (Allen).

Definition of a heuristic

Heuristics are strategies derived from experience with similar problems, using readily accessible, though loosely applicable, information to control [problem solving](https://en.wikipedia.org/wiki/Problem_solving) in human beings, machines, and abstract issues. A heuristicis any approach to problem solving, learning, or discovery that employs a practical method not guaranteed to be optimal or perfect, but sufficient for the immediate goals. Where finding an optimal solution is impossible or impractical, heuristic methods can be used to speed up the process of finding a satisfactory solution. Heuristics can be mental shortcuts that ease the cognitive load of making a decision. Examples of this method include using a [rule of thumb](https://en.wikipedia.org/wiki/Rule_of_thumb), an [educated guess](https://en.wikipedia.org/wiki/Ansatz), an intuitive judgment, stereotyping, [profiling](https://en.wikipedia.org/wiki/Profiling_(computer_programming)), or [common sense](https://en.wikipedia.org/wiki/Common_sense).

# References

Allen, James. James Dow Allen Computer Programmer . 30 11 2016 <http://fabpedigree.com/james/index.htm>.

Connect Four Perfect Solver. 30 11 2016 <http://connect4.gamesolver.org/?pos=>.

Norvig, Stuart Russell and Peter. Artificial Intelligence: A Modern Approach. Upper Saddle River: Prentice Hall, n.d.

Pearce, Adam. Connect 4 AI: How it Works. 30 11 2016 <http://roadtolarissa.com/connect-4-ai-how-it-works/>.

Tromp, John. John's Connect Four Playground. 30 11 2016 <http://tromp.github.io/c4/c4.html>.

[Pearl, Judea](https://en.wikipedia.org/wiki/Judea_Pearl) (1983). *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. New York, Addison-Wesley, p. vii. [ISBN 978-0-201-05594-8](https://en.wikipedia.org/wiki/Special:BookSources/9780201055948)