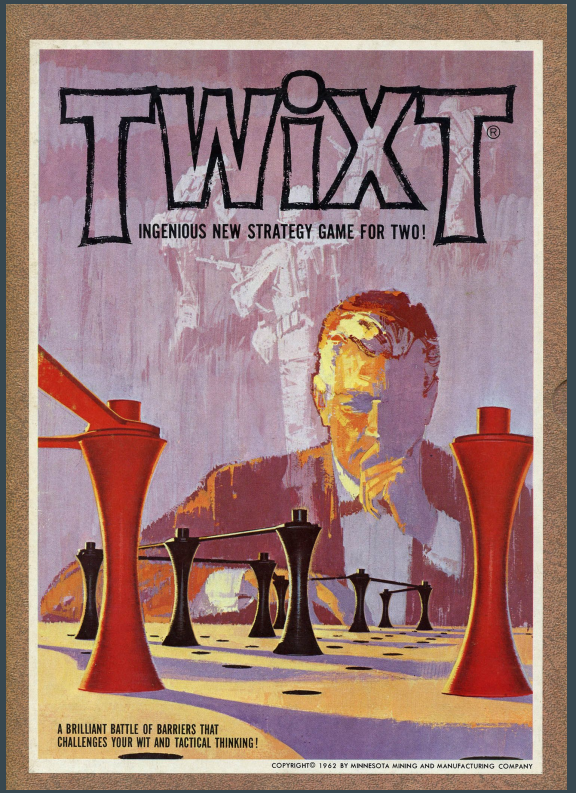
TwixT Alive

by Patrick Walsh and Jonathon Gibson

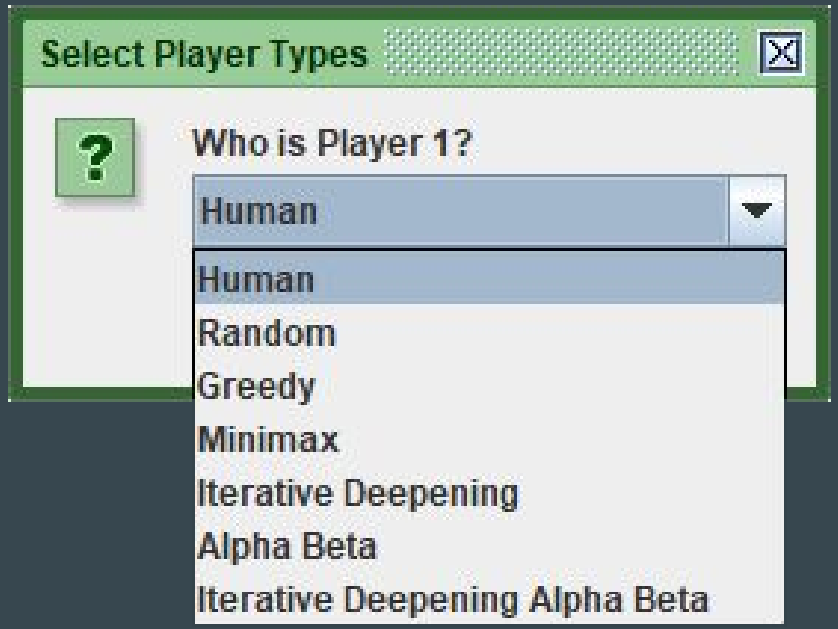
TwixT

- Two player board game
- Objective: Connect your two sides of the board with a path of dots that are a 2x1 ‘L’ distance away, paths cannot overlap
- Typically board is 24x24, with lots of variants having a smaller board size
- Highly strategic game with an annual world championship tournament
- Mediocre AI for 1-player games online



AI Opponents

- Random
- Greedy
- Minimax
- Iterative Deepening
- Alpha-Beta Pruning

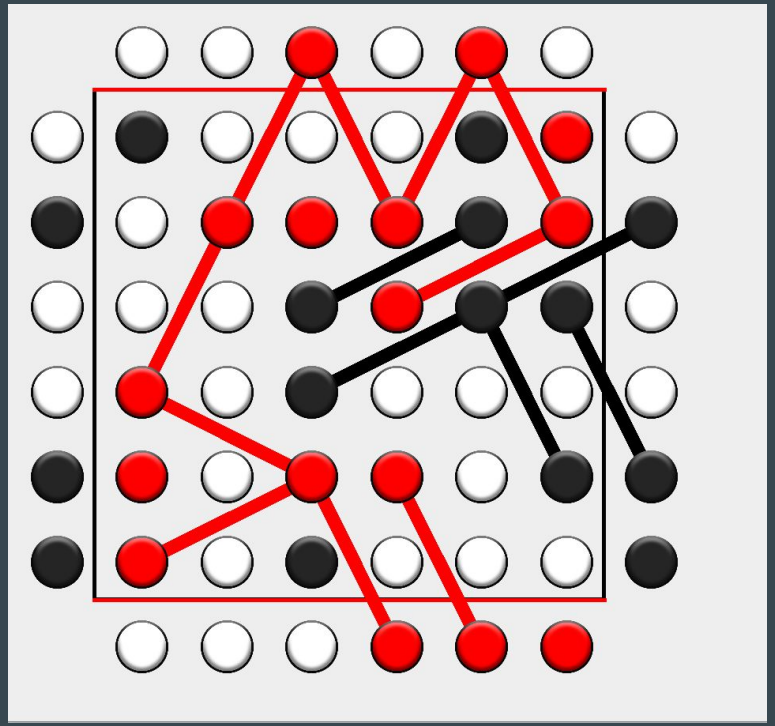


Random

The simplest opponent.

- Chooses a dot at random from all possible locations
- Repeats until that location is valid, i.e. not taken already and is not on the opponent's edge.
- Makes that move

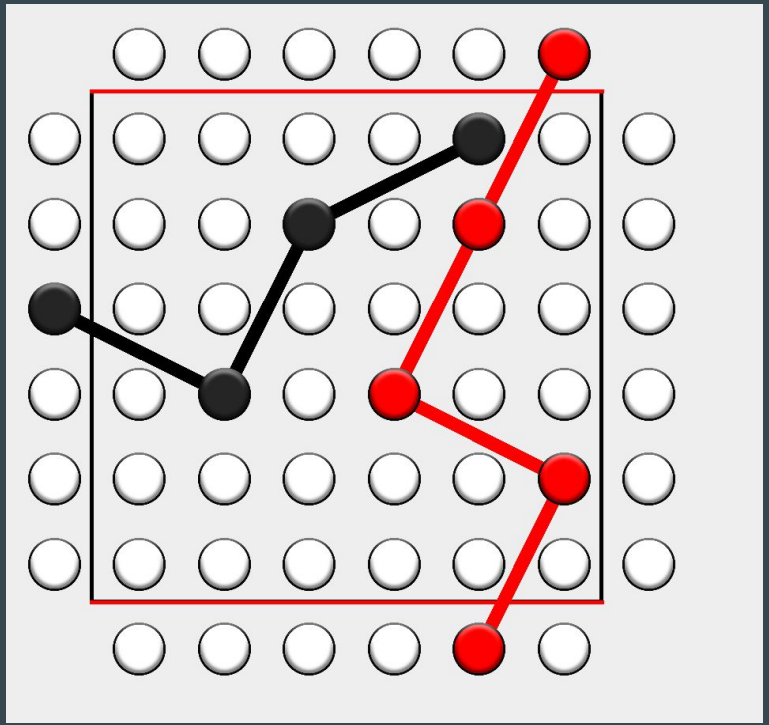
Doesn't make progress towards winning...



Greedy

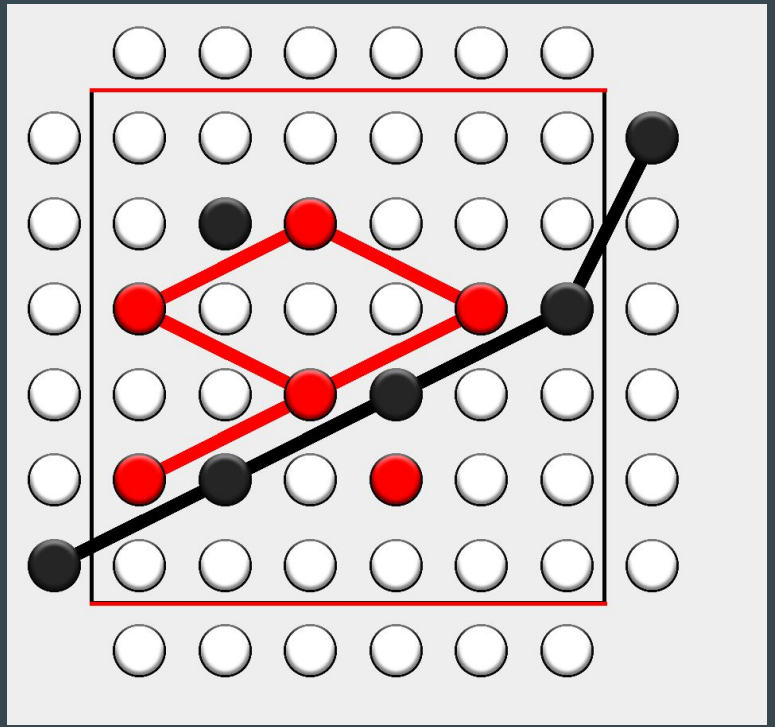
- Uses a board evaluation function to make progress towards winning
- Checks all possible location to see
  - f(x) = What is the shortest number of dots I could place to connect my sides?
  - g(x) = What is the shortest number of dots my opponent could place to connect their sides?
- Chooses the move that maximizes the opponent's "path length", while minimizing its own, i.e. the maximum of the function g(x) - f(x).
- Other heuristics and randomness

But it doesn't consider opponent's moves...



Minimax

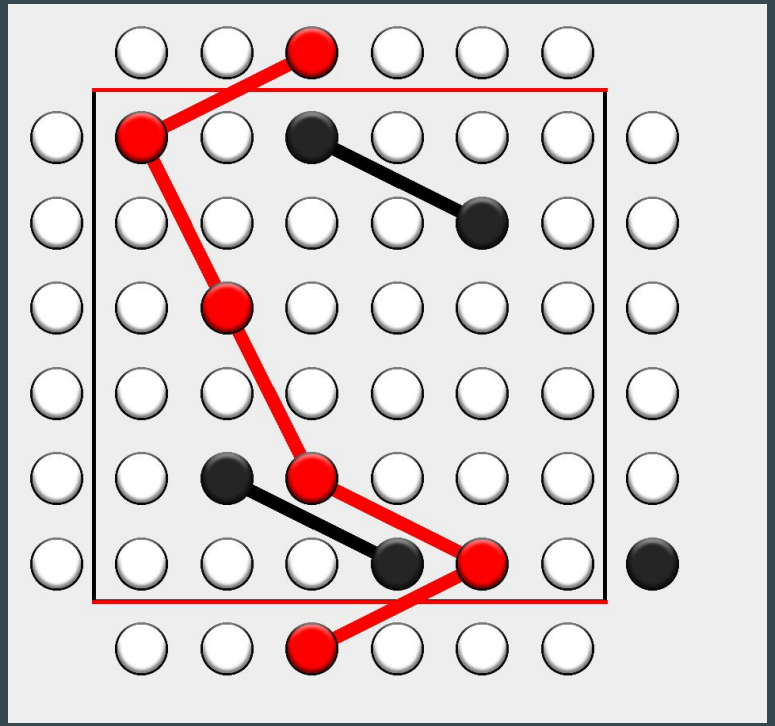
- Simulates its own moves and its opponent's moves down to a given depth using our evaluation function
- Will terminate if it finds a winning path



Doesn't allow for a set timeout or increased depth later in the game once the search space gets smaller...

Minimax Iterative Deepening

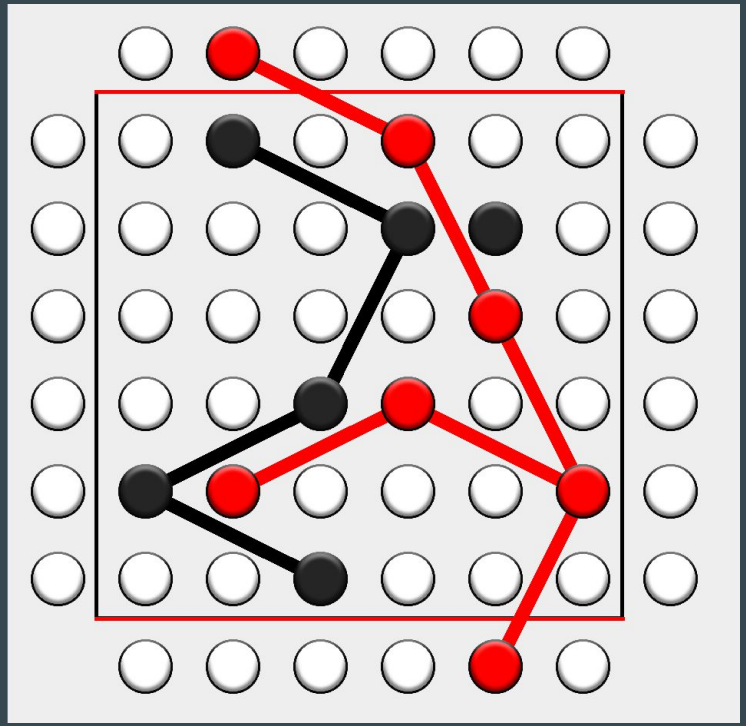
- Performs Minimax at incremental depths
- Allows us to cut off the AI's turn after a given amount of time
- Time cutoff allows for greater depth searches when board begins to fill up
- Can also limit by depth (for testing)



Very slow, in most cases limited to depth 3 in a reasonable amount of time...

Alpha-Beta

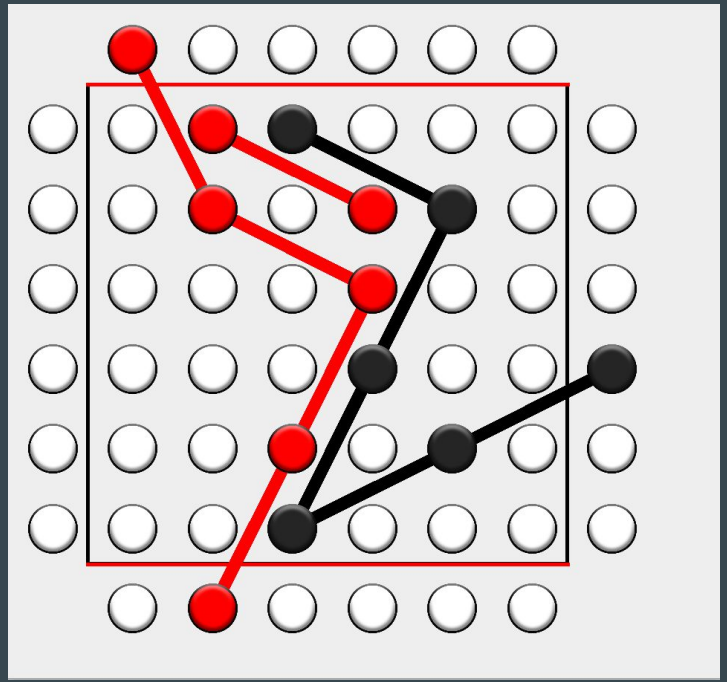
- Uses Minimax with pruning to not explore moves that won't happen
- Almost doubles the depth the AI can get in a reasonable amount of time, now around 5 instead of 3



But has same problem as Minimax, can't specify a timeout time...

Alpha-Beta Iterative Deepening

- Runs Alpha-Beta but at incrementing depth
- Allows a timeout to be set
- Gives the next depth of Alpha-Beta a starting heuristic to help increase pruning amount
  - Use previous depth's best move as the move to search first at higher depth
- Heuristic search start with Alpha-Beta pruning dramatically increases runtime



Runtime (theoretical)

NxN board

Heuristic function: O(N^2*log(N^2))

Minimax: O(b^d*h) = O((N^2)^(d+1)*log(N^2))

Alpha-Beta (Random order): O((N^2)^((3d/4)+1)*log(N^2))

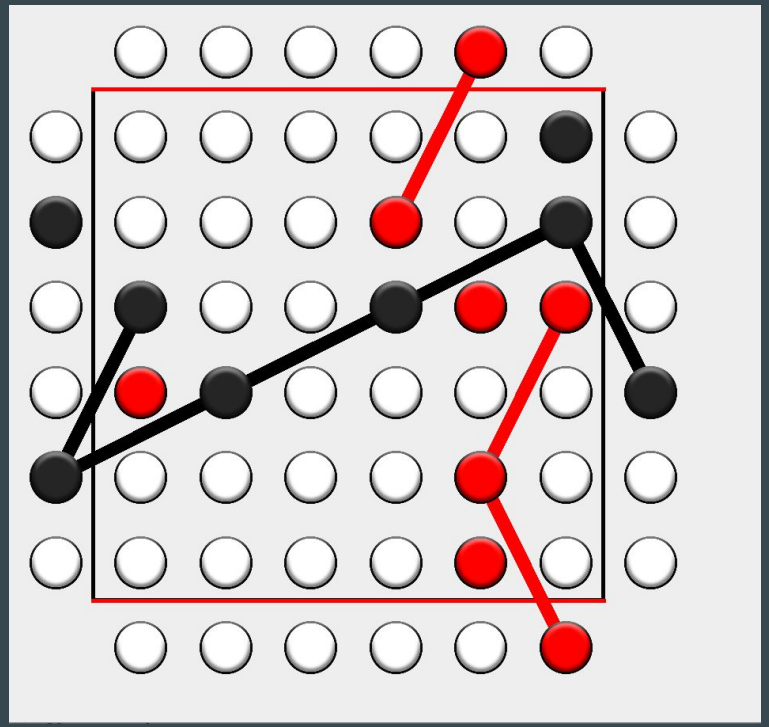
Alpha-Beta (Best order): O((N^2)^((d/2)+1)*log(N^2))

Evaluations

	Depth 1	Depth 2	Depth 3	Depth 4	Depth 5
Minimax	263	765	14478	>200000	>200000
Iterative Deepening Minimax	275	700	11440	>200000	>200000
Alpha-Beta	279	457	4334	30433	>200000
Iterative Deepening Alpha-Beta	271	434	1618	21488	>200000
Alpha-Beta Best First Approximation (w Iterative Deepening)	268	410	1604	12250	173398

Table 1. Time comparison of each AI playing itself on an 8x8 Board. Times are given in milliseconds, and represent the time taken for an entire game to be played until completion. The average of 3 trials were used for each, and a hard cutoff of 200 seconds was used.

Comparisons



Depth 4 (Black) v. Depth 2 (Red)

Vs Human?