

Jogo

Realizamos a implementação do jogo através da linguagem **JavaScript**. O jogo pode ser executado pelo arquivo `fontes/html/ligue4.html`. É interessante mencionar que testamos o jogo apenas nos navegadores **Google Chrome/Chromium** e provavelmente ocorrerá problemas na interface gráfica se for utilizado outro navegador.

A ordem dos jogadores é determinada de forma aleatória. Por isso, se você quiser que um determinado jogador seja o primeiro (**Humano** ou **Computador**) atualize a página até que ele seja sorteado como primeiro (ele ficará marcado em negrito :-).

O jogador Computador venceu.

Última jogada: **coluna 5, linha 5.**

Nodos processados: **323**
Nodos podados: **134**
Nodos folhas processados: **155**
Total de nodos processados: **301526**
Total de nodos podados: **67515**
Total de nodos folhas processados: **213261**

Minimax
Minimax com poda
Minimax com heurística A e poda
Minimax com heurística B e poda
☒ Minimax com heurística C e poda

Profundidade: **Iniciar IA**

Também foram colocados alguns contadores, incluindo contador de nodos processados, nodos podados e nodos folhas processados.

Configurações e início do jogo

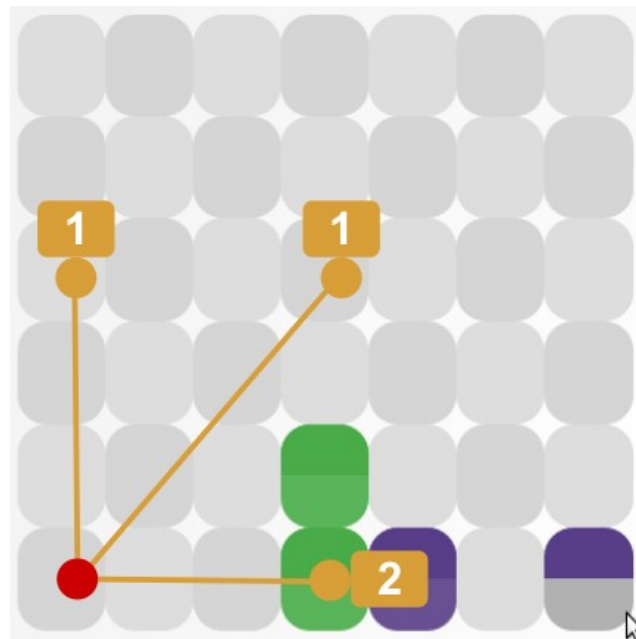
Na parte de baixo do jogo existe a possibilidade de configurar os parâmetros usados no algoritmo Minimax. É possível escolher a profundidade e a heurística. É importante perceber que após definir esses valores, você deve selecionar o botão **Iniciar**

IA. Caso o Computador seja o primeiro a jogar, também é necessário selecionar este botão para que o jogo comece. No nosso caso, uma profundidade maior que 10 já deixa a jogada do computador bastante lenta. Por isso, recomendamos que se use uma profundidade de 9 ou 10.

Heurística

Para a criação das heurísticas primeiro desenvolvemos uma heurística base e depois criamos versões alternativas para esta heurística base. No arquivo [recursos/documentos/apresentacao.pdf](#) existem algumas ilustrações que mostram como funciona o cálculo da heurística.

Basicamente a heurística consiste da soma de todas as pontuações aplicadas a cada uma das possibilidades de vitória que a jogada em uma determinada célula terá. Essa pontuação irá depender de quantas células já estão ocupadas.



No exemplo acima o valor da heurística para uma futura jogada na primeira coluna será 4. Isso acontece, pois a jogada na primeira coluna irá fazer com que a célula em questão possa participar de 3 sequências de vitórias diferentes. Entretanto, é importante notar que em uma das possíveis sequências de vitória já existe uma célula ocupada e, por isso, a pontuação daquela sequência será 2 ao invés de 1. Caso tivesse duas células ocupadas a pontuação da sequência seria 3 e assim por diante. O arquivo [recursos/documentos/apresentacao.pdf](#) mostra outros exemplos de uso da heurística.

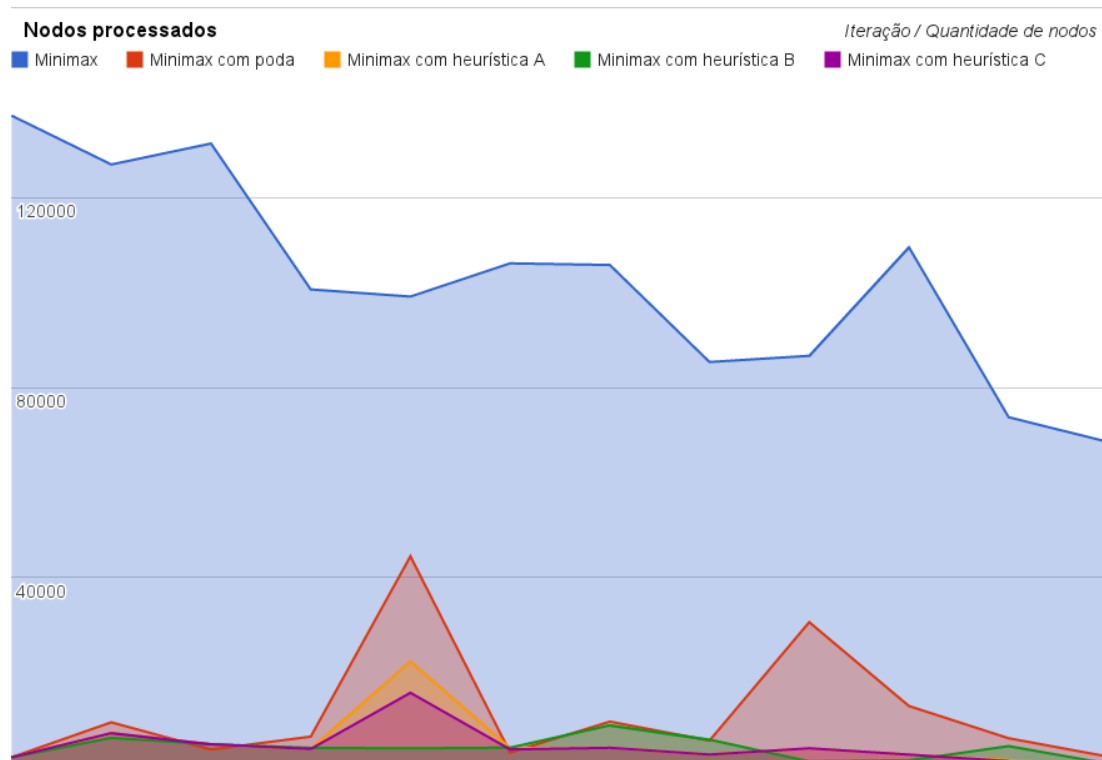
A diferença entre as três heurísticas está relacionada à qual jogador a

heurística base será aplicada. No caso da **heurística A** ele irá escolher a jogada que terá a maior pontuação de sequências vitoriosas possíveis do jogador **Computador**. Já no caso da **heurística B**, será escolhida a jogada levando em consideração a maior pontuação de sequências vitoriosas possíveis do jogador **Humano**. Ou seja, no caso da **heurística B**, o jogador **Computador** irá jogar na célula que seria escolhida pelo jogador **Humano** caso ele estivesse usando a **heurística A**. Já a **heurística C** é uma combinação das duas primeira. Assim, o **Computador** escolherá a jogada que ao mesmo tempo mais lhe dará sequências de vitórias possíveis e mais eliminará sequências de vitórias possíveis do jogador adversário.

Estatísticas

Com o jogo e as heurísticas implementadas, realizamos a coleta de estatísticas de processamento dos nodos das diferentes implementações do algoritmo Minimax. Para isso, consideramos um mesmo cenário de teste para todas as implementações:

- O jogador Humano é o primeiro;
- A profundidade foi limitada em 7 (se fosse mais iria demorar muito para processar na implementação sem poda);
- Foram utilizadas as mesmas jogadas, exceto por aquelas que fariam o jogo acabar antes da coleta completa das estatísticas;
- Foram realizadas 12 jogadas por cada jogador.



Através da análise do gráfico pode-se perceber a enorme diferença entre a quantidade de nodos processados da implementação do algoritmo Minimax sem poda em relação às outras implementações com poda.

É interessante perceber alguns picos de processamento de nodos nos algoritmos com poda. Esses picos foram ocasionados por jogadas onde o **Computador** precisou jogar na última sub-árvore da direita para não perder o jogo. Ou seja, a jogada escolhida estava na última sub-árvore mais a direita e por isso, não foi possível realizar grandes podas.

