

Leonardo Silveira Ponzoni
Lucas Gilberto Kern

RELATÓRIO SOBRE O TRABALHO DE MINIMAX

PRIMEIRA ETAPA

No nosso trabalho, nos focamos em manter a heurística mais simples possível, para que, no final, pudéssemos ampliar a profundidade da nossa árvore de busca.

Para a primeira etapa do trabalho, nosso trabalho apresentava diversas falhas que causaram uma média baixa de vitórias. Nossa primeira versão possuía dois erros fundamentais. Ele não realizava capturas e não realizava os movimentos corretos com as peças pretas. Nosso trabalho, para a primeira etapa, possuía duas heurísticas principais: Uma que levava em conta a **quantidade de peças**, dando um peso para cada uma delas, e outra que levava em consideração a **posição de cada peça** no tabuleiro.

Mesmo com essas falhas, conseguimos uma média de 35% de vitórias na primeira etapa, mostrando que estávamos no caminho certo caso conseguíssemos consertar nossas falhas principais.

PREPARAÇÃO PARA A SEGUNDA ETAPA

A preparação da segunda etapa se focou em corrigir os problemas primários da primeira etapa. Descobrimos que ambos haviam sido causados por pequenos erros na nossa codificação.

O problema da captura fora causado durante a avaliação dos estágios intermediários da nossa árvore de busca. Lá, quando avaliávamos cada possível cenário, usávamos o panorama de jogo (board) completo, ao invés do atualizado com a captura, fazendo com que a captura nunca diminuísse o valor da heurística do oponente. Procurando o erro, também fizemos otimizações de código, removendo partes irrelevantes ou ambíguas encontradas. Corrigido esse erro, o próximo era fazer com que as peças pretas se comportassem normalmente. Descobrimos que esse comportamento era causado por uma geração de duplo negativo. Ao codificar o cálculo das heurísticas, negávamos o valor, quando pretas, uma vez na peça em si e outra vez ao somar com o valor das brancas. Ao remover a negação da peça em si, resolvemos esse problema

Realizados esses ajustes, vimos um bot muito bom, porém, muito lento, devido ao nível extra da árvore de estados. Recebíamos diversas penalidades no meio do jogo, quando haviam muitas peças, ou seja, muitas possibilidades de jogada. Para corrigir esse problema adicionamos, então, uma nova heurística, a **liberdade de movimento** de cada uma das peças.

Também adicionamos uma função de *shuffle* para a lista de movimentos, para fazer com que o bot não faça sempre a mesma abertura, eliminando, assim, a previsibilidade no começo da partida.

SEGUNDA ETAPA E ANÁLISE DE ERROS

Como previsto ao final da primeira etapa, nossos resultados foram expressivamente melhores na segunda etapa. Com uma média de vitórias de, aproximadamente, 65% em 26 jogos, consideramos esse resultado uma excelente melhora. Vimos, porém, que poderíamos ter melhorado ainda mais. A principal deficiência que sentimos foi que o bot se tornou muito receoso na última jogada, isso é, ele hesitava na hora de realizar o movimento para a 8a casa do tabuleiro. Isso aconteceu por que, como havíamos uma árvore mais profunda, ele via jogadas que não aconteceriam, pois o jogo já havia acabado, não jogando, então, o peão para a vitória com medo de perdê-lo após a mesma. Outro problema que notamos é o número de penalidades recebido pelo bot quando não havia uma “matança” no começo do jogo, também devido a profundidade da nossa árvore, custando-nos, inclusive, uma derrota. Esse último erro poderia ser corrigido melhorando o peso da função de liberdade de movimento.

HEURÍSTICAS

Quantidade de Peças (Presente desde a Etapa 1)

A ideia dessa heurística vem da base do xadrez, cada peça tem seu valor, então perder uma peça não é bom, assim como capturar uma peça do oponente é. Essa é a função que diretamente dita quando o bot deve, ou não, capturar uma peça.

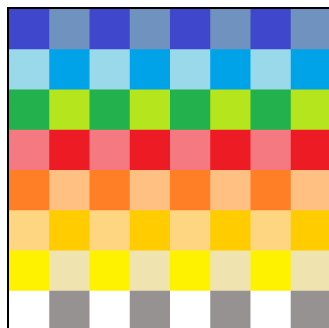
Os valores dados a função foram quanto ao valor da peça no nosso cenário de batalha de peões. Atribuímos o valor 5 para peões, 14 para os bispos e 15 para as torres. A escolha desses valores foi feita através de testes do bot contra o bot random e contra ele mesmo. A diferença entre os valores do bispo e da torre foram apenas para não ocorrer um empate na hora de capturas, dando uma pequena vantagem para a torre.

Posição de Cada Peça (Presente desde a Etapa 1)

Essa heurística trata do posicionamento das peças, preparando torres e bispos para defesa e ataque, respectivamente. Para peões, ela faz, simplesmente com que eles queriam avançar cada vez mais, sendo, assim, o principal peso para uma vitória.

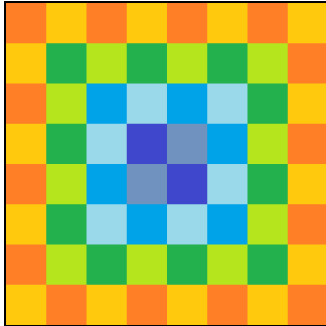
Cada peça possui uma valuação diferente dependendo da sua posição no tabuleiro:

Peão



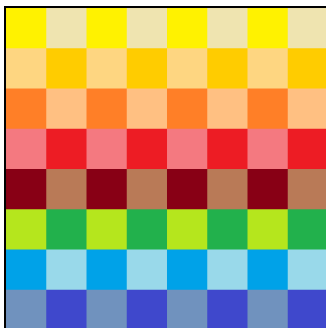
O peão possui valor baixo (amarelo) quando está perto do início, e tem seu valor aumentado progressivamente pelo tabuleiro, ganhando 2 no seu valor heurístico a cada cada movida. Além disso ele também ganha peso máximo (1.000.000) ao chegar à última casa.

Bispo



Utilizamos o bispo como nossa peça de ataque, dando maior valor para ele quando o mesmo se encontra no meio do tabuleiro, para que a cobertura do mesmo seja amplificada.

Torre



Definimos a torre como uma peça defensiva, feita para parar os peões do nosso oponente. Para isso, colocamos o valor mais alto da mesma na linha de defesa (linha 0 para brancas ou 7 para pretas). Fazendo delas uma grande muralha.

Liberdade das Peças (Presente apenas na etapa 2)

Essa heurística, como dito acima, foi criada com o objetivo de melhorar a performance do nosso bot. Ela fazia isso concedendo bônus heurísticos em função da quantidade de movimentos de cada peça. Em um primeiro momento parece contra-intuitivo, mas a ideia aqui é, quanto mais movimentos as peças podem realizar, a quantidade de peças de ambos os times é baixa. Isso funciona, pois, quando existem apenas poucas peças, a poda da árvore reduz bastante o número de estados, pois estamos com vários estados representando movimentos de uma mesma peça, fazendo com que, ao se descobrir o melhor com certa peça, todos outros movimentos da mesma possam ser, e serão, podados.