

Relatório Connect Four

Alunos: Irenio Lima Jesus de Aragão
Nildo Wilpert Júnior.

Florianópolis, abril de 2016

Objetivo

Este relatório tem como objetivo descrever os critérios e os valores que serão utilizados para implementar as funções de utilidade e heurística na implementação do algoritmo de decisão Minimax para o jogo Connect Four. Além disso, descreve a abordagem escolhida para expansão dos nodos e para identificação do nodo folha, que representa o fim do jogo, na árvore de decisão.

Expansão

Na expansão de cada nodo, para identificar as possíveis jogadas a partir de tal nodo, as peças serão avaliadas a partir da fronteira de peças do tabuleiro, ou seja, a partir das peças mais altas de cada coluna. Assim, as peças abaixo destas serão avaliadas apenas se forem do mesmo jogador e até um limite de 3 peças, desconsiderando peças que não podem fazer parte de novos agrupamentos (completar novas jogadas).

Heurística

A função heurística se trata de uma estimativa, com base em critérios pré-definidos, da proximidade entre o nodo sendo avaliado e o nodo objetivo.

Neste trabalho, os critérios utilizados para avaliação heurística serão:

- Quantidade de agrupamentos de peças do mesmo jogador com possibilidade de formar quadras (seguidas de espaços vazios), alinhadas na vertical, horizontal, diagonal esquerda ou diagonal direita, atribuindo os seguintes valores:

Quantidade de peças	Valor do agrupamento
1 (+3 vazias)	10
2 (+2 vazias)	50
3 (+1 vazias)	250

Tabela 1 - valores da função heurística

- Nível do nodo sendo avaliado (quantidade de jogadas até o estado atual). Tendo como valor o número do nível multiplicado por -5 (valoriza os níveis menores).

Considerando tais valores, a avaliação total recebe peso positivo para níveis de MAX (jogada do computador) e peso negativo (multiplicado por -1) para níveis de MIN (jogada do adversário).

Utilidade

A função de utilidade é aplicada sobre um estado final (nodo objetivo), avaliando, com base em critérios pré-definidos, o quão favorável é o estado, diferenciando finais piores e melhores. Neste trabalho, os critérios de avaliação para a função de utilidade serão os mesmos da função heurística, incluindo um valor de **5000** para agrupamentos de 4 peças do mesmo jogador (vencedor). Sendo assim, tem-se a tabela:

Quantidade de peças	Valor do agrupamento
1 (+3 vazias)	10
2 (+2 vazias)	50
3 (+1 vazias)	250
4 (vitória)	5000

Tabela 2 - valores da função utilidade

Considerando tais valores, a avaliação total recebe peso positivo para níveis de MAX (jogada do computador) e peso negativo (multiplicado por -1) para níveis de MIN (jogada do adversário).

Verificar se é nodo folha (Fim de jogo)

Para verificação do nodo folha, será identificada a última peça adicionada para se chegar ao estado atual e, então, será avaliada a vizinhança desta peça (3 peças em cada direção), verificando se formam uma quadra de peças do mesmo jogador, alinhada na vertical, horizontal ou em uma das diagonais. Caso forme, o estado é considerado um fim de jogo, sendo assim, um nodo folha.

Decisões de Projeto

Sendo o principal objetivo deste trabalho o desenvolvimento das funções heurística e de utilidade e do uso adequado das estruturas de dados, evitou-se o esforço na implementação da lógica básica do algoritmo Minimax. Para isso, foi utilizada a implementação que consta na página *Tic-tac-toe AI - Java Game Programming Case Study*¹, com as devidas adaptações para o jogo *Connect Four*.

¹ Tic-tac-toe AI - Java Game Programming Case Study. Disponível em: <http://www3.ntu.edu.sg/home/ehchua/programming/java/javagame_tictactoe_ai.html>. Acesso em: 25 maio, 2016.

Limitações

Após tentativas de implementação utilizando a linguagem Ruby, que possui melhor desempenho que o Java, encontrou-se limitações de tempo para desenvolver a regra planejada para expansão. Assim, considerando o tempo gasto e o pouco domínio da equipe sobre a linguagem Ruby, decidiu-se implementar em Java e priorizar a implementação de pontos mais importantes para este trabalho, como uma interface para o jogo e os algoritmos da função heurística e de utilidade.

Otimizações

Sendo a função de utilidade uma extensão da função heurística, o código fonte foi otimizado, centralizando a avaliação baseada na quantidade de peças. Sendo assim, para nodos não folha, avalia-se agrupamentos de até 3 peças e para nodos folha, avalia-se também as quádruplas formadas.

Outra otimização foi feita na expansão das jogadas possíveis. Estas são adicionadas a uma lista ordenada, priorizando as jogadas mais próximas da coluna central, onde há maiores possibilidades de agrupamentos, aumentando as chances de podas, visto que as jogadas com maior vantagem serão avaliadas primeiro.

Heurística

O cálculo da heurística foi dividido em 4 partes. Análises Horizontal, Vertical, Diagonal Direita e Diagonal Esquerda. Tratando o resultado como valor positivo para peças do computador e valores negativos para peças do adversário.

Foi realizada uma alteração nos valores previstos para as heurísticas. Conforme a tabela que segue:

Quantidade de peças	Valor do agrupamento
1 (+3 vazias)	10
2 (+2 vazias)	75
3 (+1 vazias)	700

Tabela 3 - novos valores da função heurística

Vertical → Para cada coluna são contadas quantas peças iguais existem a partir da peça superior da coluna. O numero de celulas vazias é calculado através da diferença entre o total de linhas e a linha da celula superior. Se o somatório entre o celulas vazias e iguais for maior que quatro é atribuída a pontuação conforme a tabela 3.

Horizontal → Para cada linha, analisam-se as peças em grupos de 4. Partindo do início da linha, conta-se quantas peças de cada e quantos espaços vazios há no grupo. Se houver peças dos dois jogadores no grupo, sabe-se que ali não há possibilidade de pontuação, pois não há possibilidade de quádrupla. De acordo com a

quantidade de peças e espaços vazios, a pontuação é acrescida (valores da *Tabela 1*). Após a avaliação, identifica-se qual o primeiro espaço em branco ou a primeira peça diferente (diferente da peça inicial do grupo) dentro das quatro. Partindo dessa peça, o novo grupo de quatro peças é criado, seguindo com a verificação e deslocamento até que as quatro últimas peças da linha sejam avaliadas. Assim, é garantido que todas as possibilidades de quádrupla foram mapeadas e atribui-se maior valor para agrupamentos que compartilham um mesmo espaço vazio (maior possibilidade de vitória).

Diagonais → São realizados agrupamentos de 4 células. Cada grupo inicia em uma célula adjacente a célula inicial anterior. Isto permite gerar de 1 a 3 grupos. Para cada um dos grupos é realizado o cálculo do número de células e atribuído um valor baseado na tabela 3. Quando há somente um grupo o valor da heurística deste é retornado. Quando existem 2 e 3 grupos é realizada uma comparação e o melhor valor de heurística é retornado.

Utilidade

O cálculo da utilidade segue o mesmo padrão do cálculo da heurística, dividido em Análises Horizontal, Vertical, Diagonal Direita e Diagonal Esquerda.

A diferença aqui é que sempre há agrupamentos de quatro peças, o que configura uma vitória por um dos jogadores e, por isso, a função utilidade é utilizada. Para essas quádruplas, atribui-se o valor constante na *Tabela 4*, induzindo a preferência por caminhos que levam à vitória do computador e evitando caminhos que levam à vitória do adversário.

Quantidade de peças	Valor do agrupamento
1 (+3 vazias)	10
2 (+2 vazias)	75
3 (+1 vazias)	700
4 (vitória)	10000

Tabela 4 - novos valores da função utilidade

Verificação de folha

A verificação do nó folha foi mantida, conforme planejamento que consta acima.

Para tal foram criados 4 métodos. Um para realizar a avaliação em cada direção.

A fim de otimizar o desempenho, e por ser a verificação menos custosa, a vertical é feita primeiramente, seguida da horizontal, diagonal direita e diagonal esquerda.

Vertical → Se a linha da última peça ≥ 3 (quarta linha) olha até 3 linhas para baixo contando quantas peças são iguais.

Horizontal → No pior caso, quando a jogada ocorrer na coluna 3 (quarta coluna, coluna do meio) a verificação será feita na linha toda. Caso a ultima coluna jogada seja menor que 3, verificação ocorre da coluna zero até a coluna jogada + 3. Caso a ultima coluna jogada seja maior que 3 a verificação ocorre da coluna jogada - 3 até a ultima coluna.

Diagonais → Primeiramente tentou-se realizar cálculos para verificar a possibilidade de vitória. E só então realizar a contagem das células. Esta abordagem gerou alguns resultados não esperados, por fim, optou-se pela utilização de dois laços de repetição para contar as peças ao lado direito e esquerdo da peça jogada. O laço para caso encontre uma peça diferente da jogada. Se total de peças contadas for maior que 3, significa uma vitória.

Estrutura de dados

Foi utilizada uma matriz simples de 7x6 para representar as colunas e linhas, respectivamente, do tabuleiro. A matriz armazena células, com um valor que indica a peça do jogador contida nesta célula, ou o vazio, e suas coordenadas.

A matriz de células está contida em uma classe chamada *Board*. Esta classe também possui como atributo a variável *lastMovement* que representa a ultima posição jogada e é utilizada na verificação de nodo folha.