

Exercício Prático 2 - BCC - 6NA

Implementação de um jogador para o jogo *conecta quatro*

Caio Castelão Sanchez, Alexandre Sierra

¹Faculdade Senac de Ciências Exatas e Tecnologia
Centro Universitário Senac
Av. Eugênio Stevaux, 82
CEP 04696-000. São Paulo

`caio.sanchez@gmail.com, alexsierra74@hotmail.com`

1. Introdução

O jogo quatro em uma linha (*Connect Four*) é um jogo de tabuleiro de duas pessoas. As jogadas são alternadas entre os jogadores. Em cada jogada cada jogador escolhe uma coluna para inserir uma bola com a cor do jogador. As bolas são inseridas em um tabuleiro vertical com sete colunas e seis linhas. O objetivo do jogo é ligar quatro bolas da mesma cor em uma linha vertical, horizontal ou diagonal - antes que seu oponente faça isso. A seguir, iremos mostrar como implementamos um jogador chamado "Timao" para o jogo *Connect Four*. Passo à passo, explicaremos a estratégia utilizada, as funções e os testes executados. O jogador foi desenvolvido na linguagem java.

2. Estratégia

A estratégia utilizada na implementação do jogador segue os passos a seguir:

1. Analisa se no tabuleiro existe alguma coluna onde o jogador ganha.
2. Analisa se no tabuleiro há alguma coluna onde é possível realizar uma ameaça dupla contra o oponente sem perder na próxima jogada.
3. Busca uma coluna onde tenha mais bolas próximas ou mais espaços vazios (máximo de 3 casas de distância em todas as direções).
4. Verifica se a escolha do passo 3 não acarreta em uma derrota no próximo movimento do oponente ou uma possível ameaça dupla do oponente, nesse caso, elimina essa escolha e tenta outra.
5. Se esgotarem todas as possibilidades, a preferência é de jogar em um lugar que o oponente não ganhe na próxima jogada. Porém, se ainda não tiver escapatória, joga na primeira coluna livre.

3. Implementação

A implementação do jogador foi feita por partes, seguindo a ordem:

1. função de abrangência
2. função `euGanhoEmUma`
3. função `ameaçaDupla`

3.1. Função de abrangência

Para iniciarmos a implementação de um jogador, começamos com uma função de abrangência que checava, apenas, uma jogada futura, a próxima jogada. Definimos pesos para saber qual a melhor jogada futura. Os pesos relacionavam as possíveis jogadas futuras com as posições das outras bolas no tabuleiro que eram adjacentes a posição atual:

- Peso 1 = espaço vazio no tabuleiro.
- Peso 90 = bola do oponente no tabuleiro.
- Peso 100 = bola minha no tabuleiro.

A função não se mostrou eficiente, porque o jogador não possuía um destino, ou alvo, para realizar o próximo movimento. Assim existia inúmeras possibilidades em que o jogador poderia ganhar na jogada posterior, mas o mesmo não jogava na posição "de vitória" no tabuleiro.

3.2. Função euGanhoEmUma

A função euGanhoEmUma foi implementada para mostrar uma meta ao jogador, a vitória. Essa função varre todas as possibilidades de jogadas futuras no tabuleiro e verifica se em alguma delas o jogador vence na próxima jogada. Caso isso ocorra, é retornada a coluna do tabuleiro da jogada vencedora.

Também aproveitamos essa função para a tática de defesa. É checado cada possível jogada do oponente com a função euGanhoEmUma e, verificado se o oponente pode nos vencer em apenas uma jogada posterior à todas as nossas possíveis jogadas. Se isso acontecer, essa jogada é banida das nossas possibilidades de movimentação.

3.3. Função ameaçaDupla

Essa função verifica se é possível ganhar em duas rodadas do oponente. Para isso, também checa se o jogador não perde na primeira jogada posterior do oponente.

A função ameaçaDupla também foi utilizada na tática defensiva. Invertendo o jogador e o oponente, beneficia o jogador, pois previne de qualquer ameaça dupla do oponente contra o jogador.

4. Testes e resultados

Os testes foram realizados em todas as fases de construção do jogador. A versão final do jogador foi testada dez mil vezes contra o oponente "Aleatório", o jogador "Timao" venceu todas. Contra o "AleatórioFocado", "Timão" venceu todas também, em dez partidas ida e volta. Quando o "Timao" jogou contra ele mesmo, houve empate. O último teste executado foi o jogador "Timao" jogando contra os seus criadores, Alexandre e Caio, e o "Timão", pasmem, venceu seus criadores.

5. Conclusão

Nosso jogador foi bem nos testes executados. Acreditamos que exista algumas melhorias que podem ser feitas, como implementar o algoritmo Min-Max com poda Alpha-Beta ou utilizar a busca quiescence, entre outras. Devido a outras atividades universitárias simultâneas, não conseguimos desenvolver e testar as melhorias com a qualidade que gostaríamos. Portanto, na versão entregue do jogador "Timao" não foi implementado o

algoritmo Min-Max.

Contudo, analisamos os relatórios dos grupos do semestre passado, que irão participar da competição, e concluimos que o nosso jogador "Timão" dificilmente ganhará dos jogadores dos grupos do semestre passado. Mas, o "Timão" ganhará fácil do jogador "baseline".