

Inteligência Artificial - Turma 8NA

Ativade Auto-Instrucional

Assunto: Criando Agentes Inteligentes para Jogos

Componentes

Caetano Motta Milorde

Luciano Bernardes Silva

Tanato Pinto Coelho Cartaxo

Estratégias e Algoritmos utilizados

Inicialmente, tentou-se construir um agente que, ao percorrer o tabuleiro, buscasse a melhor jogada e executasse a mesma em seguida, numa lógica recursiva de tentativas e testes. Dessa forma, deparou-se com problemas relacionados a estouro de pilha e baixo desempenho do agente.

Após análise do código fonte do projeto TicTacToe, disponibilizado no LAB3, baseou-se na lógica de algoritmos MiniMax, utilizando-se de uma estrutura em pilha para controle de possíveis movimentos e testes relacionados aos mesmos.

A implementação foi feita da seguinte forma:

```
privateLinkedList<Move> successors(Isola game)
```

Método do tipo lista encadeada, codificado para armazenar todos os movimentos possíveis, de acordo com a posição atual do agente.

```
publicboolean move(final Isola game)
```

Método chamado pelo servidor do jogo, para que o jogador tenha sua jogada realizada.

```
private Move minimax(Isola game)
```

Método que encabeça a lógica minimax implementada, representando a parte Max, armazenando a melhor posição até o momento.

```
privateintmaxValue(Isola game)
```

Continuando a lógica minimax, esse método trata a maximização dos menores valores para a posição atual do agente, retornando o valor máximo encontrado.

```
privateintminValue(Isola game)
```

Segundo o método anterior, esse trata a minimização dos maiores valores para a posição atual do agente, retornando o valor mínimo encontrado.

Análise dos resultados obtidos

Notou-se, como era esperado, que o tempo de resposta do agente Thanatos sem a lógica de Alpha-Beta Pruning não atingia o valor viável para o jogo (máximo 30 segundos). Uma vez a lógica implementada, esse valor foi alcançado, apesar de ter-se notado perda considerável da inteligência do agente.

AgentKeyboard x Thanatos

Devido à perda de inteligência do agente citada no parágrafo anterior, o agente desenvolvido pode ser facilmente vencido pelo agente humano.

Thanatos x Thanatos

Caso a implementação do Alpha-Beta Pruning tivesse sido feita com sucesso, o primeiro player seria o vencedor a cada execução do jogo. O contrário ocorre com o agente Thanatos.

Código fonte do agente

Arquivo Thanatos.java conforme anexado ao arquivo Luciano_Silva_AAI.zip.