Solução do 8-*Puzzle* por meio do algoritmo A*

Marco Cezar Moreira de Mattos[1], **Rômulo Manciola Meloca**[1]

[1]DACOM – Universidade Tecnológica Federal do Paraná (UTFPR)
Caixa Postal 271 – 87301-899 – Campo Mourão – PR – Brazil

{marco.cmm,rmeloca}@gmail.com

Abstract.

Resumo.

1. O Problema

O jogo 8-*Puzzle*, aos olhos humanos possui uma solução, que embora não seja trivial, bastante intuitiva, dado seu objetivo. Consiste em um tabuleiro 3x3 sobre o qual deslizam oito peças enumeradas, onde os únicos movimentos possíveis para se atingir o objetivo são aqueles permitidos pelo buraco deixado pela nona peça. O objetivo do jogo é ordenar o tabuleiro.

O problema ocorre quando não há um agente dotado de intelecto para resolver o problema, não pela complexidade das verificações feitas para atingir-se o objetivo, mas sobre quais decisões devem ser tomadas em cada estado do problema, para atingir-se a solução do problema.

O espaço dos estados do 8-*Puzzle* é 9!, portanto, sortear o próximo estado ou expandir todas as possíveis soluções jamais poderia obter a solução em tempo plausível, o primeiro porque a aleatoriedade possui a mesma probabilidade de caminhar rumo a solução quanto de caminhar no sentido oposto, o segundo porque demandaria processamento e memória difíceis de serem obtidos.

Enfim, problemas cujos espaço dos estados fogem da possibilidade viável de computação dado a complexidade do algoritmo, são resolvíveis por meio do uso de inteligência artificial, que, muito embora não forneça a melhor solução, fornece uma solução muito boa em tempo muito bom (é claro que alguns tipos de problemas são melhores resolvidos com determinados tipos de algoritmos observando-se os determinados parâmetros que o fazem comportar-se bem).

Para o 8-*Puzzle* é possível lançar-se mão desta categoria de algoritmos, contudo neste trabalho, utilizou-se o algoritmo A* que não é capaz de aprender (uma vez que armazenar resultados anteriores e observar se já foram visitados não é aprendizagem de máquina), mas que retorna um resultado muito bom em tempo viável dado sua capacidade de ignorar estados que afastam-se do objetivo e caminhar sempre rumo a ele.

2. Organização da Solução

2.1. Diagramação

2.2. Interfaces

2.3. Protocolo

3. Implementação

Implementou-se a solução utilizando a linguagem de programação Java, contando com um objeto Puzzle e abstrações para os movimentos possíveis.

A heurística utilizada foi distância de Manhattan combinada com segundo [1] refrências latex.

4. Resultados

5. Considerações Finais

6. References

Bibliographic references must be unambiguous and uniform. We recommend giving the author names references in brackets, e.g. [Knuth 1984], [Boulic and Renault 1991], and [Smith and Jones 1999].

The references must be listed using 12 point font size, with 6 points of space before each reference. The first line of each reference should not be indented, while the subsequent should be indented by 0.5 cm.

Referências

Boulic, R. and Renault, O. (1991). 3d hierarchies for animation. In Magnenat-Thalmann, N. and Thalmann, D., editors, *New Trends in Animation and Visualization*. John Wiley & Sons ltd.

Knuth, D. E. (1984). *The TEX Book*. Addison-Wesley, 15th edition.

Smith, A. and Jones, B. (1999). On the complexity of computing. In Smith-Jones, A. B., editor, *Advances in Computer Science*, pages 555–566. Publishing Press.