

Klotski / 47

Resolução do Klotski utilizando Métodos de Pesquisa em Linguagem Javascript (Tema 1/Grupo 47)

Mário Santos 201503406
FEUP
Vila Nova de Gaia, Portugal
up201503406@fe.up.pt

Mateus Pedroza 201601876
FEUP
Porto, Portugal
up201601876@fe.up.pt

Filipe Reis 201506154
FEUP
Porto, Portugal
up201506154@fe.up.pt

Este trabalho será realizado no âmbito da Unidade Curricular de Inteligência Artificial como método prático de aprendizagem das competências para desenvolvimento de algoritmos de IA. Utilizamos o jogo/puzzle “Klotski” como contexto para a implementação desses mesmos algoritmos.

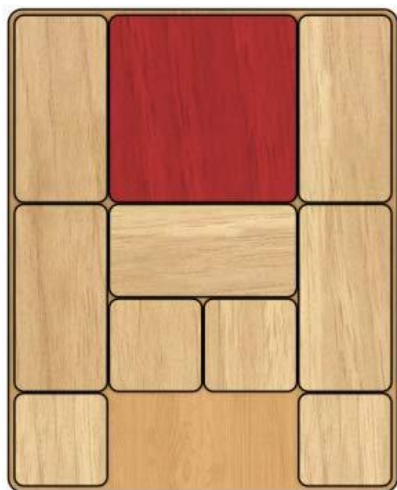
Keywords—*Inteligência Artificial, Pesquisa em largura, Pesquisa em profundidade, Aprofundamento progressivo, Pesquisa de custo uniforme, Pesquisa gulosa, Algoritmo A*.*

I. INTRODUÇÃO

O puzzle *Klotski* foi escolhido dada a sua natureza gráfica e possíveis abordagens relativamente à sua resolução através de algoritmos de IA, começando com o algoritmo BFS (Breadth-First Search), e dependendo dos resultados, possivelmente algoritmos mais eficientes. Este artigo está dividido em 5 partes, sendo que esta se trata da introdução, e as seguintes, por ordem, Descrição do Problema, Formulação do Problema, Trabalho Relacionado e Conclusões e Perspetivas de Desenvolvimento, sendo disponibilizado no fim as respetivas referências bibliográficas.

II. DESCRIÇÃO DO PROBLEMA

Klotski é puzzle de blocos deslizantes, originado no início do séc. XX. O seu país de origem é desconhecido pois existem muitas afirmações contraditórias, e vários países afirmam ser a derradeira origem do jogo.



Tabuleiro inicial mais comum

O nome refere-se a um conjunto de 10 blocos, onde o objetivo é mover um bloco específico para uma posição pré-determinada. As regras são as seguintes, tal como outros puzzles de blocos deslizantes, vários blocos são colocados num tabuleiro, geralmente com dimensões 4x5. Entre os blocos existe um bloco especial, distinguível através da sua cor vermelha e por ser o maior, que tem de ser movido para

uma área especial designada pelo tabuleiro. O jogador não pode remover blocos, e só pode deslizar os blocos horizontalmente ou verticalmente. O objetivo é resolver o puzzle, com o mínimo número de jogadas, ou a mínima quantidade de tempo.



Exemplos de tabuleiros possíveis

III. FORMULAÇÃO DO PROBLEMA

Estamos a considerar como estado do nosso jogo, a posição das peças numa matriz que representa o tabuleiro.

Os blocos normais têm dimensões de 1x1, 2x1 ou 1x2 e bloco vermelho tem dimensões de 2x2

O estado inicial tem o canto superior esquerdo do bloco vermelho na primeira linha, na segunda coluna. Os restantes blocos estão no resto do tabuleiro sem pré-disposição específica.

O teste objetivo é que o canto superior esquerdo do bloco vermelho se encontre na quarta linha, na segunda coluna

Os operadores são o movimento das peças para as posições vazias, havendo um número de sucessores igual ao número de movimentos de blocos possíveis. O nome do operador é movimentar peça para direção, a pré-condição é que cada elemento da matriz para onde o bloco se quer mover, não esteja ocupado, o efeito é a reescrita do estado de jogo num novo estado e o custo de cada movimento é igual.

IV. TRABALHO RELACIONADO

Descrevendo outros trabalhos semelhantes que os estudantes tenham encontrado na sua pesquisa e códigos fonte úteis para a realização do trabalho (devendo ser incluídas referências aos trabalhos e códigos fontes na seção de referências bibliográficas).

O primeiro exemplo trata-se de um *puzzle solver* para o jogo *Klotski* desenvolvido pela Universidade de San Diego. Utiliza o algoritmo de *breadth-first* a pesquisar a solução e utiliza de uma ferramenta para medir a eficiência da resolução com posições diferentes de peças.[2]

O segundo exemplo trata-se de uma biblioteca em javascript para resolver um jogo Klotski utilizando o algoritmo BFS (Breadth-first search). Há possibilidades de medir a eficiência e distintos padrões de configuração.[3]

O terceiro e último exemplo é menos relevante porém é um algoritmo de resolução do puzzle Klotski pelo método BFS(Breadth-first search), feito em Java e foi utilizado modificações para obter a melhor performance possível no algoritmo.[4]

V. CONCLUSÕES E PERSPETIVAS DE DESENVOLVIMENTO

Com este trabalho prevemos conseguir entender que método de pesquisa obterá melhores resultados na procura de uma solução com o mínimo de passos possíveis,

VI. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Jogo Klotski na wikipedia a 18 de março de 2019
- [2] <https://en.wikipedia.org/wiki/Klotski> 18 de março de 2019
- [3] <https://github.com/twyatt/klotski-solver> 18 de março de 2019
- [4] <https://github.com/jeantimex/klotski> 18 de março de 2019
- [5] <https://gist.github.com/mpd89/5283698> 18 de março de 2019