

Resolução do jogo Cohesion utilizando Métodos de Pesquisa em Linguagem Python (Tema 4/ Grupo 33)

João Alves (201605236)
Faculdade de Engenharia da
Universidade do Porto
Porto, Portugal
up201605236@fe.up.pt

Amadeu Pereira (201605646)
Faculdade de Engenharia da
Universidade do Porto
Porto, Portugal
up201605646@fe.up.pt

Nuno Lopes (201605337)
Faculdade de Engenharia da
Universidade do Porto
Porto, Portugal
up201605337@fe.up.pt

Resumo- Este artigo contém a descrição de um trabalho cujo objetivo é implementar o jogo Cohesion e resolver diferentes níveis do mesmo, utilizando diferentes métodos de pesquisa.

Os métodos de pesquisa aplicados são comparados a diferentes níveis, dando mais importância, dentro destes, ao tempo de execução, qualidade da solução e ao número de movimentos executados.

Keywords: *Inteligência Artificial, Pesquisa, Algoritmo A*, Cohesion, Pathfinding*

I. INTRODUÇÃO

A pesquisa é a técnica universal de resolução de problemas em Inteligência Artificial. Em problemas de IA, a sequência de passos necessários para a solução de um problema não é conhecida à priori, mas deve ser determinada por uma exploração sistemática de tentativa e erro de alternativas. Os problemas abordados pelos algoritmos de pesquisa da IA enquadram-se em três grupos gerais: *Single Agent Pathfinding Problems, Multiplayer Games and Constraint-Satisfaction Problems*.

Este artigo está estruturado da seguinte forma: a seção II descreve de forma pouco pormenorizada mas facilmente perceptível o problema/jogo em questão; a seção III formaliza o problema como um problema de pesquisa de IA; a seção IV relaciona o trabalho com outros problemas semelhantes; fazemos uma avaliação do trabalho desenvolvido até ao momento e das partes em processo de construção na Seção V.

II. DESCRIÇÃO DO PROBLEMA

O jogo Cohesion é um puzzle baseado no jogo “15 Puzzle” onde os quadrados têm várias cores. As peças podem ser jogadas para qualquer lado desde que não haja nenhuma peça de outra cor que impeça.

Se dois quadrados/peças da mesma cor se tocam eles unem-se num só permanentemente, originando assim uma peça nova.

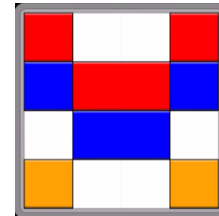


Ilustração 1 - Exemplo de um nível do puzzle.

O objetivo deste puzzle é juntar todas as peças da mesma cor.

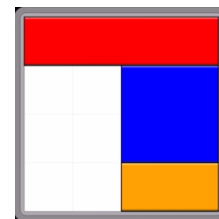


Ilustração 2 - Exemplo de um estado objetivo do puzzle.

III. FORMULAÇÃO DO PROBLEMA

Representação do Estado: Matriz 4x4 em que cada elemento pertence a $\{0, \dots, N\}$, sendo N o número total de cores distintas existentes no jogo. Quando um elemento é 0 significa que não existe nenhuma peça (cor) nesta posição.

Estado Inicial: O estado inicial depende do nível em questão. Consiste numa distribuição de várias cores pela matriz, garantindo que as peças da mesma cor não se encontram todas juntas.

Estado Objetivo: As peças da mesma cor estão todas adjacentes umas às outras.

Operadores:

Quando 2 peças da mesma cor se tocam formam um bloco permanente, tendo este conjunto de peças da mesma cor de se mover em conjunto.

- **esquerda(pos):**

Pré-Condição: todas as posições à esquerda do bloco da peça selecionada(pos) tem que ser uma posição vazia.

Efeito: o bloco anda uma posição para a esquerda.

Custo: 1.

- direita(pos):

Pré-Condição: todas as posições à direita do bloco da peça selecionada(pos) tem que ser uma posição vazia.

Efeito: o bloco anda uma posição para a direita.

Custo: 1.

- cima(pos):

Pré-Condição: todas as posições em cima do bloco da peça selecionada(pos) tem que ser uma posição vazia.

Efeito: o bloco anda uma posição para cima.

Custo: 1.

- baixo(pos):

Pré-Condição: todas as posições em baixo do bloco da peça selecionada(pos) tem que ser uma posição vazia.

Efeito: o bloco anda uma posição para baixo.

Custo: 1.

Custo da solução: Cada movimento custa 1, logo o custo da solução é o número total de movimentos.

IV. TRABALHO RELACIONADO

Não conseguimos encontrar nenhum código-fonte do jogo Cohesion, logo como tal decidimos basearmo-nos em implementações do jogo “15 Puzzle” uma vez que tem certas semelhanças ao nosso puzzle. Sendo assim encontramos a implementação de uma pessoa chamada Milan Pecov [2] onde

implementa algoritmo A* e pesquisa em largura para o jogo “15 Puzzle”.

Também para a implementação deste jogo vamos utilizar o código disponível no link [1] pois é possível encontrar a implementação do jogo em diversas linguagens de programação.

V. CONCLUSÕES E PERSPETIVAS DE DESENVOLVIMENTO

Primeiramente, acreditamos que o nosso conhecimento sobre métodos de pesquisa está a ser aprofundado de forma gradual, mostrando-se assim o grupo satisfeito com a escolha do problema. De facto, este tipo de problemas relacionados com pathfinding são geralmente desafiadores, e pelo facto de estarem associados a jogos fez com que o interesse do grupo aumentasse.

Neste momento ainda só desenvolvemos uma parte lógica do jogo, faltando ainda acabar esta mesma parte e iniciar a utilização dos métodos de pesquisa. É evidente que ainda há muito trabalho pela frente, mas o grupo acredita que o trabalho está no bom caminho.

Pretendemos testar vários métodos de pesquisa para assim comparar diferentes aspetos computacionais.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Rosetta Code, “15 Puzzle Game”, last updated March 2019, [online], available at: https://rosettacode.org/wiki/15_Puzzle_Game, consulted on March 2019.
- [2] Milan Pecov, “15-puzzle”, 2013, [online], available at: <https://github.com/MilanPecov/15-Puzzle-Solvers>, consulted on March 2019.