

# **Trabalho 1 – Fase 1**

FCUP 2021/2022

Inteligência Artificial

Daniela Coelho up202004799

Luna Figueiredo up202007414

Patrícia Rodrigues up202008466

A linguagem utilizada para a implementação das estratégias de busca foi C++ pois estamos familiarizadas com técnicas suportadas pela linguagem C, mas também com as estruturas de dados já feitas em C++.

A estrutura de dados que utilizamos para representar o tabuleiro do jogo dos 15 foi uma `struct Tab` (Tabuleiro), composta por uma matriz de inteiros `m[4][4]`, 4 por 4, e mais duas variáveis do tipo `int`. Estas variáveis servem para armazenar a linha (`zx`) e a coluna (`zy`) referentes à posição do 0 no tabuleiro. Deste modo, cada tabuleiro irá ocupar 72 bytes. Esta estrutura de dados serve para guardar o tabuleiro inicial e o tabuleiro a que queremos chegar, guardando em ambos os casos a posição em que o 0 se encontra.

De maneira a analisar se um tabuleiro tem solução, começamos por definir uma função `uniArray`, do tipo `void`. Esta recebe como argumento uma matriz de inteiros 4 por 4 (`m`) e armazena os seus valores no array unidimensional `inicial_uni` declarado como variável global.

De seguida, criamos a função `solvable`, do tipo `bool`. Esta analisa o array `inicial_uni` e a posição do 0 de um tabuleiro `t` (passado como argumento). Começa por contar o número de inversões no array. Posteriormente, aplica a fórmula para determinar a resolubilidade: se o número de inversões for ímpar e o 0 estiver numa linha par ou se o número de inversões for par e o 0 estiver numa linha ímpar, a função retorna `true` (`t` é solucionável); retorna `false` nos restantes casos, indicando que `t` não é solucionável.

Esta função será útil para comparar a configuração inicial com a configuração final, isto é, verificar se da configuração inicial se consegue chegar à configuração final.

Criamos ainda uma função `moveZero` que nos permite mover a posição do 0 no tabuleiro. Esta recebe como argumento um tabuleiro `t` (`Tab t`) e um inteiro, que serve para representar a direção na qual o 0 se irá mover. Assim, através de um `switch` conseguimos mover o 0 para cima, para baixo, para a esquerda e para a direita (`case 1, 2, 3, 4`, respetivamente), manipulando os índices da matriz. Finalmente, a função retorna um novo tabuleiro com as respetivas mudanças, ou seja, com o 0 na nova posição.