

Inteligência Artificial

# Match The Tiles

Grupo 39

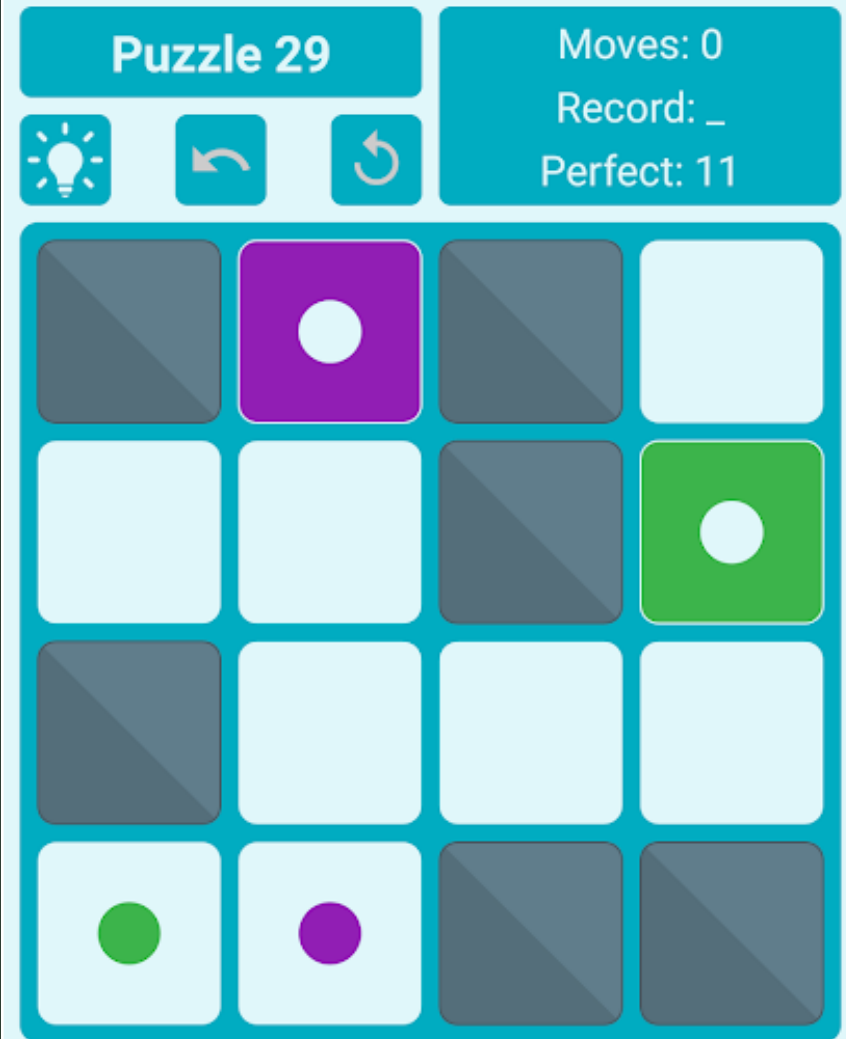
Mariana Ramos – up201806869

Pedro Ferreira – up201806506

Pedro Ponte – up201809694

# Especificação do Projeto

- O jogo consiste na existência de um tabuleiro contendo diferentes tipos de “tiles”, correspondendo cada tile a uma célula do tabuleiro.
- Existem três tipos de tiles no jogo, onde cada um apresenta uma cor diferente que os relaciona com o seu objetivo no jogo. Tiles preenchidas com uma cor escura correspondem a paredes, tiles com um ponto no centro correspondem a tiles objetivo e tiles pintadas com um círculo no centro correspondem a tiles jogáveis.
- O objetivo do jogo é colocar as tiles jogáveis nas posições onde se encontram as tiles objetivo com a cor correspondente. Para tal, o jogador pode deslocar as peças para cima, baixo, esquerda e direita. Os movimentos são sincronizados, portanto, deve-se usar tiles fixos existentes para criar espaços entre os tiles e resolver o puzzle.



# Referências e Trabalho Relacionado

- [Link para a página do jogo na Google Play;](#)
- [Exemplo para desenvolvimento gráfico;](#)
- [Exemplo de desenvolvimento de algoritmos de pesquisa](#)
- Slides das aulas teóricas e teórico-práticas para desenvolvimento dos diferentes algoritmos.

# Formulação do Problema como um problema de pesquisa

## ▪ Estado de Representação

O tabuleiro de jogo é representado por uma matriz B quadrangular com Y colunas e Y linhas,  $4 \leq Y \leq 6$ . Os valores de cada célula têm os seguintes valores:

- 'X', no caso de ser uma parede,
- 'B', 'G', 'O', 'P', 'R' ou 'Y' no caso de ser uma célula jogável
- 'BF', 'GF', 'OF', 'PF', 'RF' ou 'YF' no caso de ser uma célula destino
- '-' no caso de ser uma célula vazia.

## ▪ Heurística:

$f(n) = g(n) + h(n)$ , onde  $g(n)$  é o custo do caminho desde o estado inicial até ao nó n e  $h(n)$  é uma função heurística que estima o custo do caminho mais barato desde n até ao estado objetivo.

## ▪ Estado Inicial

Matriz B com estado inicial desejado. Por exemplo:

```
B = [ [ 'X', 'X', '-', 'X' ],  
      [ '-', 'X', '-', 'IP' ],  
      [ 'FP', 'X', 'IP', 'X' ],  
      [ '-', '-', '-', 'FP' ] ]
```

## ▪ Estado Objetivo

Matriz F sem estados finais (FZ) sozinhos. Por exemplo:

```
B = [ [ 'X', 'X', '-', 'X' ],  
      [ '-', 'X', '-', '-' ],  
      [ 'IPFP', 'X', '-', 'X' ],  
      [ '-', '-', '-', 'IPFP' ] ]
```

- Operadores:

Nome	Pré-condição	Efeitos	Custo
Up	$\exists IZ, B[ IZx, IZy-1 ] \notin \{ "X", "IZ" \}$ $\wedge IZy > 0$	$\forall IZ,$ <i>While</i> ( $B[ IZx-1, IZy ] \notin \{ "X", "IY" \}$ ) <i>do</i> $B[ IZx, IZy-1 ] = IZ$	1
Down	$\exists IZ, B[ IZx, IZy + 1 ] \notin \{ "X", "IZ" \}$ $\wedge IZy < B.size$	$\forall IZ,$ <i>While</i> ( $B[ IZx+1, IZy ] \notin \{ "X", "IY" \}$ ) <i>do</i> $B[ IZx, IZy+1 ] = IZ$	1
Left	$\exists IZ, B[ IZx-1, IZy ] \notin \{ "X", "IZ" \} \wedge$ $IZx > 0$	$\forall IZ,$ <i>While</i> ( $B[ IZx, IZy-1 ] \notin \{ "X", "IY" \}$ ) <i>do</i> $B[ IZx-1, IZy ] = IZ$	1
Right	$\exists IZ, B[ IZx+1, IZy ] \notin \{ "X", "IZ" \}$ $\wedge IZx < B.size$	$\forall IZ,$ <i>While</i> ( $B[ IZx, IZy+1 ] \notin \{ "X", "IY" \}$ ) <i>do</i> $B[ IZx+1, IZy ] = IZ$	1

Legenda: IZ - peça que se move IY - outra peça X - parede



# Detalhes de Implementação

## Linguagem de programação

- Python, recorrendo ao pacote *pygame* para representação da interface do jogo.

## Ambiente de desenvolvimento

- VSCode / Spyder.

## Estruturas de dados

- Listas, para representar os tabuleiros de jogo;
- *Nodes* e *Graphs*.

## Estrutura de ficheiros:

- A figura ao lado representa a estrutura dos ficheiros com diferentes matrizes para diferentes níveis com os quais podemos testar o jogo;

## Trabalho já implementado

- Representação gráfica do tabuleiro;
- Verificação dos movimentos válidos;
- Modo de jogo individual;
- Implementação do algoritmo BFS.

```
level1 = [  
    ['X', '-', '-', '-'],  
    ['-', '-', '-', '-'],  
    ['-', '-', 'X', 'IB'],  
    ['FB', '-', '-', '-']  
]  
  
level4 = [  
    ['X', 'IR', '-', '-', '-'],  
    ['FP', '-', '-', '-', 'X'],  
    ['X', '-', 'X', 'X', '-'],  
    ['X', '-', '-', 'IP', 'X'],  
    ['IG', '-', 'X', 'X', 'X']  
]
```