

# Trabalho final IAA - Treasure Hunt

Miguel Correia 57609

Carlos Rodrigues 60630

9 de janeiro de 2025



# Conteúdo

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Introdução</b>   | <b>3</b> |
| <b>2</b> | <b>Sumário do Tema</b>  | <b>3</b> |
| <b>3</b> | <b>Design e Arquitetura</b>   | <b>3</b> |
| 3.1      | Probabilidade do Estado Inicial da Localização do Tesouro . . . . . | 3        |
| 3.2      | Bayesian Network . . . . .  | 3        |
| 3.2.1    | Detection Mode . . . . .  | 3        |
| 3.2.2    | Dig Mode . . . . .  | 4        |
| 3.3      | Conditional Probability Tables . . . . .                            | 4        |
| 3.3.1    | Detect Mode . . . . .   | 5        |
| 3.4      | Dig Mode . . . . .  | 6        |
| <b>4</b> | <b>Desafios e Reflexões</b>   | <b>6</b> |

# 1 Introdução

Este projeto foi realizado no âmbito da cadeira de Inteligência Artificial Aplicada do Mestrado em Engenharia Informática, sob a tutela do professor António Anjos. Tem como objetivo de aplicar todo o conhecimento adquirido durante as aulas.

## 2 Sumário do Tema

O objetivo do projeto é implementar um jogo interativo, em Python, que usa os mecanismos de deteção probabilística numa grelha de  $m \times n$ . Os jogadores têm que localizar o tesouro escondido usando o modo detetor de forma estratégica, já que, por cada uso, o jogador recebe uma penalização. É claro que, se os jogadores forem confiantes, podem usar o modo de escavação para encontrar o tesouro, mas esse modo é uma jogada garantida para finalizar o jogo. Se for com sucesso, o jogador ganha e recebe todos os pontos que lhes foi atribuído, se não for, o jogador perde.

## 3 Design e Arquitetura

Durante a implementação do nosso jogo, damos a opção aos jogadores de escolher o tamanho da grelha de jogo, mas para o relatório o tamanho será  $m = n = 3$ .

### 3.1 Probabilidade do Estado Inicial da Localização do Tesouro

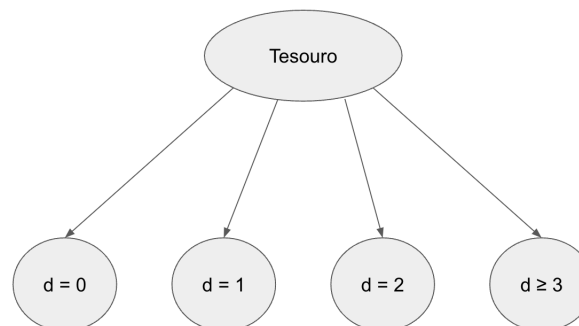
Podemos assumir que a distribuição da probabilidade irá ser uniforme, cada célula da grelha de tamanho  $m \times n$  terão todos a mesma probabilidade de obter o tesouro no início do jogo e é demonstrado da seguinte forma:

$$P(T = \text{célula}) = \frac{1}{m \times n}$$

### 3.2 Bayesian Network

#### 3.2.1 Detection Mode

- **Nós:**
  - Localização do tesouro (T): Representa a localização atual do tesouro na grelha.
  - Sinais de deteção ( $D_1, D_2, D_3, \dots, D_d$ ): Representa os sinais que o jogador recebe por cada vez que insere coordenadas e é definido pela distância  $d$ .
- **Arestas:** A localização do tesouro afeta cada sinal de deteção ( $T \rightarrow D_d$ )



### 3.2.2 Dig Mode

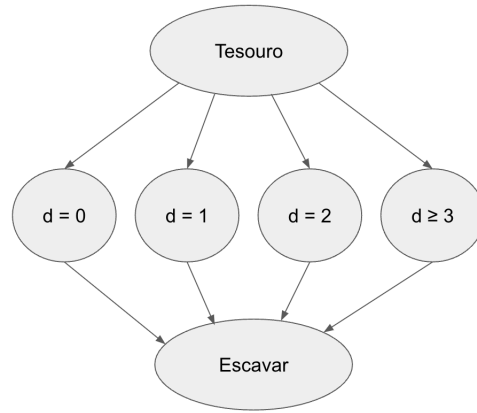
- Nós:

- Localização do tesouro (T): Representa a localização atual do tesouro na grelha.
- Sinais de deteção ( $D_1, D_2, D_3, \dots, D_d$ ): Representa os sinais que o jogador recebe por cada vez que insere coordenadas e é definido pela distância  $d$ .
- Local de escavação (G): Representa as coordenadas inseridas pelo jogador para escavar o tesouro.

- Arestas:

- $T \rightarrow D_d$ : A localização do tesouro afeta cada sinal de deteção.
- $D_d \rightarrow G$ : A escolha das coordenadas de escavação é afetada pelos sinais de deteção.

A localização do tesouro afeta cada sinal de deteção ( $T \rightarrow D_d$ )



### 3.3 Conditional Probability Tables

Primeiro, o jogo irá começar com todas as células com a probabilidades iguais, representado nesta CPT:

| Célula | P(T=célula) |
|--------|-------------|
| (0, 0) | 0,111       |
| (0, 1) | 0,111       |
| (0, 2) | 0,111       |
| (1, 0) | 0,111       |
| (1, 1) | 0,111       |
| (1, 2) | 0,111       |
| (2, 0) | 0,111       |
| (2, 1) | 0,111       |
| (2, 2) | 0,111       |

### 3.3.1 Detect Mode

- $P(T|D_d)$

| d (distância) | S (sinal) | $P(T D_d)$ |
|---------------|-----------|------------|
| 0             | ++++      | 0.8        |
| 0             | +++       | 0.1        |
| 0             | ++        | 0.07       |
| 0             | +         | 0.03       |
| 1             | ++++      | 0.08       |
| 1             | +++       | 0.8        |
| 1             | ++        | 0.08       |
| 1             | +         | 0.04       |
| 2             | ++++      | 0.04       |
| 2             | +++       | 0.08       |
| 2             | ++        | 0.8        |
| 2             | +         | 0.08       |
| ≥3            | ++++      | 0.03       |
| ≥3            | +++       | 0.07       |
| ≥3            | ++        | 0.1        |
| ≥3            | +         | 0.8        |

- Estado seguinte

Após o jogador usar o modo de detecção na célula (1, 0), é aplicado a seguinte expressão:

- **Distância de Manhattan:**  $d = |i - i_d| + |j - j_d|$ , em que  $(i_d, j_d)$  é a célula detetada.
- **Atualização das probabilidades:** Para cada célula (i, j) na grelha  $G(i, j) = L(i, j) \times \text{Prior}(i, j)$ , em que L representa a probabilidade do sinal mais provável na célula (i, j) e Prior representa a probabilidade do estado anterior, L é calculado assim:

$$L = \text{signal\_table}[\min(d, 3)][S]$$

em que S representa o sinal obtido após detecção.

- **Probabilidade total:**  $\text{prob\_total} = \sum_{i=1}^i \sum_{j=1}^j G(i, j)$
- **Normalização das probabilidades:**  $G(i, j) = \frac{G(i, j)}{\text{prob\_total}}$

E assim ficamos com a CPT da seguinte forma:

| Célula | P(T=célula) |
|--------|-------------|
| (0, 0) | 0,066       |
| (0, 1) | 0,033       |
| (0, 2) | 0,025       |
| (1, 0) | 0,656       |
| (1, 1) | 0,066       |
| (1, 2) | 0,033       |
| (2, 0) | 0,066       |
| (2, 1) | 0,033       |
| (2, 2) | 0,025       |

### 3.4 Dig Mode

Com base no que foi executado no Detect Mode, a escolha da célula é feita com base nos dados obtidos. Assim, podemos apresentar da seguinte forma:

| Célula a cavar | S (sinal) | P(T=célula) | Com tesouro |
|----------------|-----------|-------------|-------------|
| (0, 0)         | +++       | 0,066       | Alta        |
| (0, 1)         | ++        | 0,033       | Média       |
| (0, 2)         | +         | 0,025       | Baixa       |
| (1, 0)         | ++++      | 0,656       | Muito Alta  |
| (1, 1)         | +++       | 0,066       | Alta        |
| (1, 2)         | ++        | 0,033       | Média       |
| (2, 0)         | +++       | 0,066       | Alta        |
| (2, 1)         | ++        | 0,033       | Média       |
| (2, 2)         | +         | 0,025       | Baixa       |

## 4 Desafios e Reflexões

Durante o desenvolvimento do projeto, achamos mais desafiante a construção da Bayesian Network e a Conditional Probabilities Table (CPT), daí termos que nos basear bastante no que implementamos no código. Não nos sentimos limitados pois o Python fornece uma grande variedade de bibliotecas para que possamos usufruir e mesmo que não fosse o caso, teríamos outras formas de chegar aos resultados pretendidos. Cada membro contribuiu no código, sendo que o Update das probabilidades, inicialização da grelha, distância de manhattan, cálculo da probabilidades posteriores usando as anteriores foi implementado por Miguel Correia e a parte visual do jogo, o dig mode, a Bayesian Network e a CPT's para Detect e Dig Mode foram implementados por Carlos Rodrigues. Em geral, cada membro contribuiu de forma igual para o projeto.