

# Universidade da Beira Interior

## Departamento de Informática



**Departamento de  
Informática**

### **Projeto de IA**

Elaborado por:

**Ana Carolina Silva, João Fraga**

Orientador:

**Professor Doutor Luís Alexandre**

2 de Janeiro de 2020

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
1.1	Resumo . . . . .	2
1.2	Motivação . . . . .	2
1.3	Objetivos . . . . .	2
<b>2</b>	<b>Implementação</b>	<b>3</b>
2.1	Introdução . . . . .	3
2.2	Dependências . . . . .	3
2.3	Representação da Informação . . . . .	4
2.4	Resposta às questões . . . . .	5
2.4.1	Quantos quartos não estão ocupados? . . . . .	5
2.4.2	Quantas suítes foram encontradas até agora? . . . . .	5
2.4.3	É mais provável encontrar pessoas nos corredores ou dentro dos quartos? . . . . .	5
2.4.4	Se quisesse encontrar um computador, a que tipo de quarto me devo dirigir? . . . . .	5
2.4.5	Qual o número do quarto <i>single</i> mais próximo? . . . . .	6
2.4.6	Como ir do quarto atual até ao elevador? . . . . .	6
2.4.7	Quantos livros estima encontrar nos próximos 2 minutos? . . . . .	6
2.4.8	Qual a probabilidade de encontrar uma mesa num quarto sem livros e que tem pelo menos uma cadeira? . . . . .	6
<b>3</b>	<b>Reflexão Crítica</b>	<b>7</b>
3.1	Introdução . . . . .	7
3.2	Divisão do Trabalho . . . . .	7
3.3	Problemas Encontrados . . . . .	7
3.4	Análise Crítica . . . . .	8
<b>4</b>	<b>Conclusões e Trabalho Futuro</b>	<b>9</b>
4.1	Conclusão . . . . .	9
4.2	Trabalho Futuro . . . . .	9

# 1 Introdução

## 1.1 Resumo

Este trabalho consistiu no desenvolvimento de um agente robótico inteligente, que navega por um mundo, onde efetua observações. Através destas observações este é capaz de responder a questões que lhe são colocadas.

Para atingir esse objectivo foram utilizados os conhecimentos e algoritmos adquiridos na unidade curricular de Inteligência Artificial [5], nomeadamente: Algoritmos de Grafos e conhecimentos de probabilidades.

## 1.2 Motivação

Este projeto foi proposto e desenvolvido no âmbito da unidade curricular de Inteligência Artificial, com a finalidade de praticar e consolidar algumas das técnicas e conhecimentos adquiridos ao longo do semestre.

## 1.3 Objetivos

Este projeto tem como objetivo a implementação de um agente inteligente. Este deverá ser capaz de responder a questões colocadas pelo utilizador através do processamento de informação recolhida do ambiente.

## 2 Implementação

### 2.1 Introdução

Neste capítulo são apresentadas as bibliotecas utilizadas, as estruturas de dados escolhidas bem como todos os detalhes de implementação das respostas às questões.

### 2.2 Dependências

Ao longo da realização do trabalho foram utilizadas algumas bibliotecas *built-in* e *third-party* para auxílio à implementação da solução.

- Bibliotecas *built-in* do Python:
  1. *math*
  2. *time*
- Bibliotecas *third-party*:
  1. *networkx* [1]

## 2.3 Representação da Informação

Durante a navegação do agente por parte do utilizador é registada informação, recolhida pelos sensores.

Foi necessário escolher estruturas de dados para representar esta informação de forma a facilitar a sua consulta e resposta às questões. As estruturas de dados escolhidas foram as seguintes:

- **Grafo:** Foi utilizado para representar todos os quartos visitados. Uma *edge* entre dois quartos indica uma porta e o seu *weight* corresponde à distância euclidiana entre o centro dos dois quartos.
- **Dicionários:** Foram utilizados varios dicionários com o objectivo de obter consultas rápidas
  - **Dicionário de Objetos Encontrados:** devolve uma lista de objectos encontrados num determinado quarto de uma certa categoria.
    - \* **key:** (*room*, *category*)
    - \* **value:** *object\_name*
  - **Dicionário de Tipos de Quarto:** devolve qual o tipo de um determinado quarto
    - \* **key:** *room*
    - \* **value:** *type\_of\_room*
- **Listas:** Foram utilizadas várias listas de forma a representar dados mais simples
  - coordenadas centrais dos quartos
  - numero dos corredores
  - ...

## 2.4 Resposta às questões

Nesta secção é explicado de forma sucinta a implementação da solução para cada uma das perguntas sugeridas.

### 2.4.1 Quantos quartos não estão ocupados?

Fazendo uso do grafo e dicionário de objetos encontrados (secção 2.3), iterou-se sobre todos os quartos visitados e contou-se em quantos destes se tinham observado pessoas.

Subtraindo os dois valores obtem-se o número de quartos não visitados.

### 2.4.2 Quantas suites foram encontradas até agora?

Esta questão foi respondida recorrendo ao dicionário de tipos de quartos (secção 2.3). Um quarto poderia ser classificado como suite em duas situações distintas:

1. Se fosse observada uma cama nesse quarto e esse já se encontrasse ligado a outro quarto, que não um corredor, no Grafo.
2. Se fosse adicionada uma *edge* entre dois quartos (não corredores) e um desses quartos tivesse uma cama.

### 2.4.3 É mais provável encontrar pessoas nos corredores ou dentro dos quartos?

Considera-se que seria mais provável encontrar pessoas nos corredores se a proporção de corredor com pessoas fosse maior que a proporção de quartos com pessoas, e vice-versa.

Para isto efetuou-se a contagem de corredores visitados (*visited\_corridors*), corredores com pessoas (*corridors\_with\_people*), quartos visitados (*visited\_rooms*), quartos com pessoas (*rooms\_with\_people*).

Se:

$$\frac{people\_corridors}{visited\_corridors} > \frac{people\_rooms}{visited\_rooms} \quad (1)$$

Então será mais provável encontrar pessoas em corredores, caso contrário, será mais provável encontrar pessoas em quartos.

### 2.4.4 Se quisesse encontrar um computador, a que tipo de quarto me devo dirigir?

Fazendo uso do dicionário de tipos de quartos e do dicionário de objetos observados por quarto, contamos o número de quartos de cada tipo e o número de quartos com computador por tipo de quarto.

Calcula-se seguidamente a proporção de quartos com computador para cada tipo de quarto e é apresentado ao utilizador o que apresenta maior proporção.

### 2.4.5 Qual o número do quarto *single* mais próximo?

Para responder a esta questão começou-se por obter uma lista com todos os quartos *single* encontrados até ao momento.

Utilizou-se a função *shortest\_path\_length* [3] da biblioteca *networkx* [1] para calcular a distância da posição atual a cada um destes quartos.

Por fim verificou-se qual destes se encontra a menor distância.

### 2.4.6 Como ir do quarto atual até ao elevador?

Nesta questão fez-se uso da função *shortest\_path* [2] da biblioteca *networkx* [1] para obter uma lista de quartos, desde o quarto atual (origem), até ao elevador (destino).

Por fim, foram apresentadas de forma intuitiva as direções ao utilizador.

### 2.4.7 Quantos livros estima encontrar nos próximos 2 minutos?

Para responder a esta questão foram tidos em conta dois parâmetros:

1. Número de quartos visitados por segundo (*rooms\_sec*)
2. Número de livros observados por quarto (*books\_room*)

Tendo em conta estes dois parâmetros, a estimativa do número de livros a encontrar nos próximos 2 minutos é dada por:

$$prediction = (2 * 60) * rooms\_sec * books\_room \quad (2)$$

É também tido em conta o número de quartos não visitados (*remaining\_rooms*). Caso o número de quartos que é possível visitar em 2 minutos seja superior ao número de quartos não visitados, usa-se a equação:

$$prediction = remaining\_rooms * books\_room \quad (3)$$

### 2.4.8 Qual a probabilidade de encontrar uma mesa num quarto sem livros e que tem pelo menos uma cadeira?

A probabilidade pedida é dada por: [4]

$$P(T|\neg B \cap C) = \frac{P(T \cap \neg B \cap C)}{P(\neg B \cap C)} \quad (4)$$

- *T* - Ter mesa
- *B* - Ter livro
- *C* - Ter cadeira

Para efetuar este cálculo contou-se o número de quartos que não tinham livros nem cadeiras, e quantos destes tinham mesas.

## 3 Reflexao Critica

### 3.1 Introdução

Neste capítulo é analisado e discutido todo o trabalho realizado, de que forma este foi dividido pelos elementos do grupo e quais os problemas e dificuldades que surgiram.

### 3.2 Divisão do Trabalho

As decisões relativas à escolha das estruturas de dados utilizadas foi feita em conjunto por ambos os elementos do grupo de forma a evitar a duplicação de informação.

Após ter sido decidido como representar a informação obtida, dividimos as questões em 2 grupos considerados de dificuldade equivalente e cada um dos elementos do grupo implementou o seu conjunto respectivo de questões.

1. Set *A*: Resolvido por João Fraga
  - questão 1 (Secção 2.4.1)
  - questão 3 (Secção 2.4.3)
  - questão 6 (Secção 2.4.6)
  - questão 7 (Secção 2.4.7)
2. Set *B*: Resolvido por Ana Carolina Silva
  - questão 2 (Secção 2.4.2)
  - questão 4 (Secção 2.4.4)
  - questão 5 (Secção 2.4.5)
  - questão 8 (Secção 2.4.8)

Apesar desta divisão, todas as soluções foram discutidas por ambos os elementos do grupo de forma a confirmar a sua correção.

### 3.3 Problemas Encontrados

A navegação do agente era trabalhosa o que dificultava o teste das soluções, para além disso o simulador não era muito estável e requeria várias tentativas antes de funcionar correctamente.

Tendo em conta que o utilizador é responsável pela navegação do agente, por vezes poderá haver objetos que não são observados, levando à má classificação de quartos e erros na resposta às questões em geral.

Também devido à forma como o simulador está implementado, o agente consegue por vezes observar objectos através de paredes o que uma vez mais pode levar à má classificação de quartos e erros na resposta às questões em geral.



### **3.4 Análise Crítica**

Apesar de terem sido atingidos todos os objectivos propostos, não foi feito uso de algumas das técnicas mais sofisticadas de Inteligência Artificial que teriam sido interessantes de explorar.

## 4 Conclusões e Trabalho Futuro

### 4.1 Conclusão

Os objetivos estabelecidos foram cumpridos, sendo que foi dada resposta a todas as questões propostas. Este trabalho permitiu a implementação de uma aplicação prática dos conhecimentos adquiridos na unidade curricular de Inteligência Artificial.

### 4.2 Trabalho Futuro

Futuramente seria interessante desenvolver uma forma de o agente ser movido autonomamente, não controlado pelo utilizador, e continuar a expandir o número de questões ao qual este é capaz de responder.

## Referências

- [1] Networkx documentation, 2019. "<https://networkx.github.io/documentation/latest/>". Acedido em 28 de Dezembro de 2019.
- [2] `shortest_path()` documentation, 2019. "[https://networkx.github.io/documentation/stable/reference/algorithms/generated/networkx.algorithms.shortest\\_paths.generic.shortest\\_path.html#networkx.algorithms.shortest\\_paths.generic.shortest\\_path](https://networkx.github.io/documentation/stable/reference/algorithms/generated/networkx.algorithms.shortest_paths.generic.shortest_path.html#networkx.algorithms.shortest_paths.generic.shortest_path)". Acedido em 28 de Dezembro de 2019.
- [3] `shortest_path_lenght()` documentation, 2019. "[https://networkx.github.io/documentation/stable/reference/algorithms/generated/networkx.algorithms.shortest\\_paths.generic.shortest\\_path\\_length.html#networkx.algorithms.shortest\\_paths.generic.shortest\\_path\\_length](https://networkx.github.io/documentation/stable/reference/algorithms/generated/networkx.algorithms.shortest_paths.generic.shortest_path_length.html#networkx.algorithms.shortest_paths.generic.shortest_path_length)". Acedido em 28 de Dezembro de 2019.
- [4] Luís Alexandre. Inteligência artificial. Lecture Notes, 2019.
- [5] Stuart J. Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Pearson, 2016.