

Departamento de Informática

Universidade da Beira Interior



Departamento de Informática

PROJETO INTELIGÊNCIA ARTIFICIAL

Relatório do Projeto da Unidade Curricular

Elaborado por:

Carlos Martins – 41968

Rui Ferreira - 41064

Orientador: **Professor Doutor Luís Andrade**

Janeiro 2021

Índice

1 – Introdução	2
1.1 Resumo.....	2
1.2 Motivação.....	2
1.3 Objetivos	2
2 - Implementação.....	3
2.1 Introdução	3
2.2 Dependências.....	3
2.3 Representação da Informação	3
2.4 Funções Auxiliares.....	4
2.5 Respostas às questões.....	6
2.5.1 Pergunta 1: Qual foi a penúltima pessoa que viste?.....	6
2.5.2 Pergunta 2: Em que tipo de sala estás agora?.....	6
2.5.3 Pergunta 6: Quanto tempo achas que falta até ficares sem bateria?.....	7
2.5.4 Pergunta 7: Qual a probabilidade de encontrar um livro numa divisão, se já encontraste uma cadeira?.....	8
2.5.5 Pergunta 8: Se encontrares um enfermeiro numa divisão, qual é a probabilidade de estar lá um doente?.....	9
3 - Reflexão Crítica	10
3.1 Introdução.....	10
3.2 Divisão do Trabalho.....	10
3.3 Problemas Encontrados.....	10
3.4 Análise Crítica	11
4 - Conclusão	11
4.1 Conclusão.....	11
4.2 Trabalho Futuro	12

1 – Introdução

1.1 Resumo

Este trabalho visa o desenvolvimento da parte de inteligência artificial de um robô, que se desloca sob o nosso controlo, num mundo virtual, que corresponde a um piso de um hospital. Ao longo dos seus movimentos, elabora observações ao meio envolvente, a partir das quais adquire informações necessárias para responder às questões que lhe são colocadas.

1.2 Motivação

A principal motivação encontrada para o desenvolvimento deste processo centraliza-se na aprendizagem de novos conceitos e solidificação dos já adquiridos sobre Inteligência Artificial, os quais podem vir a ser bastante úteis quer para nossa vida universitária quer para a futura vida profissional.

1.3 Objetivos

Este projeto tem como principal objetivo a implementação da parte de inteligência num agente robótico, o qual deverá ser capaz de responder a questões colocadas pelo utilizador através da recolha de informação no ambiente.

2 - Implementação

2.1 Introdução

Neste capítulo abordamos as bibliotecas a que recorremos para o desenvolvimento do projeto, como também todos os detalhes necessário à implementação das funções necessárias para responder às questões colocadas.

2.2 Dependências

No decorrer do desenvolvimento do trabalho prático foi necessário recorrer a algumas bibliotecas *built-in* do Python.

Bibliotecas *built-in* do Python:

1. Time

2.3 Representação da Informação

- **Listas:** Recorreu-se à utilização de listas, de modo a conseguir armazenar informação necessária ao desenvolvimento do projeto.

○ Lista Divisões Hospital

```
room_number = [
    ((1, 85, 30, 565, 135)),
    ((2, 30, 90, 85, 330)),
    ((3, 565, 30, 635, 330)),
    ((4, 30, 330, 770, 410)),
    ((5, 85, 135, 235, 330)),
    ((6, 280, 135, 385, 330)),
    ((7, 430, 135, 565, 330)),
    ((8, 635, 30, 770, 85)),
    ((9, 635, 130, 770, 185)),
    ((10, 635, 230, 770, 330)),
    ((11, 30, 410, 235, 570)),
    ((12, 280, 410, 385, 570)),
    ((13, 430, 410, 570, 570)),
    ((14, 615, 410, 770, 570))]
```

Figura 1- Representação das divisões do hospital

De modo a definir as várias divisões do piso do hospital, definiu-se desde logo esta lista. Relativamente aos elementos que a constituem são tuplos do tipo (númeroDivisão, x1, y1, x2, y2), onde o númeroDivisão representa o número da sala, (x1, x2) os limites máximos no eixo do x e (y1, y2) os limites máximos no eixo do y para a divisão considerada.

- **Lista Itens**

```
items_list = []
```

Figura 2 – Lista Itens

Com o objetivo de guardar as várias informações recolhidas pelo robot, ao longo dos seus movimentos pelo hospital, decidimos criar esta lista. É constituído por tuplos do tipo (númeroDivisão, categoriaObjeto, nomeObjeto), onde o númeroDivisão corresponde à divisão onde o robot o encontrou, o categoriaObjeto representa a categoria do objeto encontrado e o nomeObjeto o nome atribuído a esse objeto. Aquando a deteção de um novo objeto por parte deste robot, este vai ser inserido na lista.

- **Lista Tempo**

```
times_list = []
```

Figura 3 – Lista Tempo

Esta lista tem como objetivo guardar o tempo que os últimos 3 pontos percentuais da bateria gastos demoraram a ser gastos.

2.4 Funções Auxiliares

- **Função Auxiliar *find_item(item_name)***

```
def find_item(item_name):
    for i in range(len(items_list)):
        if (item_name == items_list[i][2]):
            return True
    return False
```

Figura 4 - Função *find_item(item_name)*

Recorreu-se à criação desta função auxiliar com o objetivo de verificar se o objeto já se encontra na lista *items_list*, que representa todos os objetos encontrados pelo robô até ao momento.

- Função Auxiliar *add_new_object(objetos)*

```
def add_new_object(objetos):
    global current_room

    for item in range(len(objetos)):
        item_cate, item_name = tuple(objetos[item].split("_", 1))
        if find_item(item_name) == False:
            items_list.append((current_room, item_cate, item_name))
```

Figura 5 – Função *add_new_object(objetos)*

O objetivo desta função é adicionar o objeto detetado pelos movimentos do robot caso ainda não pertença à lista. A verificação que é feita através da função *find_item(item_name)*. Usa-se o método *split()* para separar cada string presente na lista “objetos” de modo a obter a categoria e nome separadamente.

- Função Auxiliar *current_position(x,y)*

```
def current_position(x,y):
    for room in room_number:
        if x >= room[1] and x <= room[3] and y >= room[2] and y <= room[4]:
            return room[0]
    return -1
```

Figura 6 - Função *current_position(x,y)*

Esta função tem como objetivo devolver o número da divisão do hospital onde se encontra o robot, iterando sobre a lista de divisões previamente criada.

- Função Auxiliar *time_remaining(bateria,time_used)*

```
def time_remaining(bateria, time_used):

    global times_list

    times_list.append(time_used)
    len_times_list = len(times_list)
    if len_times_list > 3:
        times_list.pop(0)

    average_times = sum(times_list) / len_times_list
    result = average_times * (101 - bateria + 1)

    return result
```

Figura 7 - *time_remaining(bateria,time_used)*

Quando esta função é invocada são passados dois parâmetros, *bateria* e *time_used*. O *time_used* representa a duração em segundos do último ponto percentual gasto relativo à bateria. Este é adicionado à lista *times_list*. A variável *average_times* representa a média dos últimos três valores recebidos. Após o cálculo da média, o mesmo é multiplicado pela capacidade restante da bateria. Este valor é considerado o tempo que falta até a bateria se esgotar.

2.5 Respostas às questões

2.5.1 Pergunta 1: Qual foi a penúltima pessoa que viste?

```
for i in range(len(objetos)):
    aux_objetos = objetos[i].split(",")
    if any(s for s in aux_objetos if "enfermeiro" in s or "doente" in s or "medico" in s):
        if list_people[0] != aux_objetos:
            list_people[1] = list_people[0]
            list_people[0] = aux_objetos
```

Figura 8 - Resolução Questão 1

Para responder a esta questão criou-se inicialmente uma lista de tamanho dois (*list_people* []), que guarda os últimos dois objetos que o robô deteta. Esses objetos tanto podem pertencer à categoria enfermeiro, como doente ou médico, pois todas essas categorias representam uma pessoa. Por fim, a penúltima pessoa a ser vista pelo robot corresponde à que se encontra na última posição da *list_people* [].

2.5.2 Pergunta 2: Em que tipo de sala estás agora?

```
def room_type(room):
    list_aux = []
    if room in [1, 2, 3, 4]:
        return "Corredor"

    for i in items_list:
        if i[0] == room:
            list_aux.append(i)

    mesa = 0
    cadeira = 0
    for i in list_aux:
        if i[1] == "cama":
            list_aux.clear()
            return "Quarto"
        if i[1] == "mesa":
            mesa += 1
        if i[1] == "cadeira":
            cadeira += 1

    if mesa >= 1 and cadeira >= 1:
        list_aux.clear()
        return "Sala Enfermeiro"
    elif cadeira >= 2:
        list_aux.clear()
        return "Sala de espera"
    else:
        list_aux.clear()
        return "Divisao"

def resp2():
    global position_aux
    #determina e guarda o tipo de divisao
    room_name[0] = room_type(current_position(position_aux[0], position_aux[1]))
    print(room_name[0])
```

Figura 9 - Utilização da função *room_type(room)*

Figura 10 – Resolução Questão 2

Com vista à resolução da segunda questão, desenvolveu-se uma função para classificar o tipo de divisão onde se encontra o robô. Inicialmente, verificamos se o mesmo se encontra no corredor. Caso não se encontre, iremos correr a lista de itens detetados até ao momento pelo robô, e quando existir um elemento da divisão onde atualmente o agente robótico se encontra adiciona-se esse objeto a uma lista auxiliar criada localmente.

Seguidamente volta-se a correr a lista auxiliar, em adição cria-se também um contador para contar o número de mesas e cadeiras.

Caso o número de mesas e cadeiras seja superior a 1 e não haja camas o robot encontra-se numa Sala de Enfermeiro, caso haja pelo menos uma cama o robot encontra-se num quarto, por fim se não existir camas nem mesas e existir mais de duas cadeiras o robot encontra-se numa sala de espera. Para o caso de nenhuma das condições anteriores ser verificada, o robot envia uma mensagem onde transmite que não reconheceu nenhuma divisão.

De referir também que aquando a classificação de cada sala é concluída, a lista criada localmente com os objetos volta a estar vazia.

2.5.3 Pergunta 6: Quanto tempo achas que falta até ficares sem bateria?

```
if int(battery_aux) != int(bateria):
    time_pass = time.time() - start_time
    battery_lost = 100 - int(bateria)
    battery_remaining = time_remaining(battery_lost, time_pass)
    battery_aux = bateria
    start_time = time.time()
```

Figura 11 – Resolução Questão 6

Sempre que a percentagem da bateria altera, determinamos o tempo que esse ponto percentual durou. Calculamos também o valor de bateria gasto até ao momento. Com esses valores invocamos a função auxiliar *time_remaining()* referida anteriormente. No final atualizamos a variável “start_time” para o tempo atual.

2.5.4 Pergunta 7: Qual a probabilidade de encontrar um livro numa divisão, se já encontraste uma cadeira?

A probabilidade pedida é dada por:

$$P(A / B) = \frac{P(A \cap B)}{P(B)}$$

- A – Ter livro
- B – Ter cadeira

```
def resp7():
    room_with_chair = 0
    room_with_chair_and_book = 0
    rooms_visited = 0

    for room in range(5, 15):
        aux_visited = False
        aux_chair = 0
        aux_book = 0

        for item in items_list:
            if item[0] == room:
                if item[1] == "cadeira":
                    aux_chair += 1
                if item[1] == "livro":
                    aux_book += 1
                aux_visited = True

        if aux_visited == True:
            rooms_visited += 1
        if aux_chair >= 1:
            room_with_chair += 1
        if aux_chair >= 1 and aux_book >= 1:
            room_with_chair_and_book += 1

    if(rooms_visited == 0 or room_with_chair == 0):
        print("Nenhuma divisao com cadeira encontrada")
    else:
        pb_chair = room_with_chair / rooms_visited
        pb_chair_and_book = room_with_chair_and_book / rooms_visited
        res = pb_chair_and_book / pb_chair
        print("A probabilidade é %.2f." % res)
```

Figura 12 – Resolução Questão 7

Para calcular esta probabilidade é necessário contar o número de quartos que contêm cadeiras e livros, o número de quartos que contêm cadeiras, como também o número de quartos visitados. Caso não exista quartos com cadeiras, o robô apresenta uma mensagem a informar que não encontrou nenhuma divisão com cadeiras.

2.5.5 Pergunta 8: Se encontrases um enfermeiro numa divisão, qual é a probabilidade de estar lá um doente?

A probabilidade pedida é dada por:

$$P(A / B) = \frac{P(A \cap B)}{P(B)}$$

- A – Ter doente
- B – Ter enfermeiro

```
def resp8():
    room_with_enf = 0
    room_with_enf_and_doente = 0
    rooms_visited = 0

    for room in range(5, 15):
        aux_visited = False
        aux_enf = 0
        aux_doente = 0

        for item in items_list:
            if item[0] == room:
                if item[1] == "enfermeiro":
                    aux_enf += 1
                if item[1] == "doente":
                    aux_doente += 1
                aux_visited = True

        if aux_visited == True:
            rooms_visited += 1
            if aux_enf >= 1:
                room_with_enf += 1
            if aux_enf >= 1 and aux_doente >= 1:
                room_with_enf_and_doente += 1

    if(rooms_visited == 0 or room_with_enf == 0):
        print("Nenhuma divisao com enfermeiro encontrada")
    else:
        pb_enf = room_with_enf / rooms_visited
        pb_enf_and_doente = room_with_enf_and_doente / rooms_visited
        res = pb_enf_and_doente / pb_enf
        print("A probabilidade é %.2f." % res)
```

Figura 14 - Resolução Questão 8

De modo a calcular esta probabilidade, efetua-se a contagem do número de quartos que contêm enfermeiros e doentes, apenas enfermeiros e o número total de quartos visitados. Na situação de não existir divisões com enfermeiros, o robô envia a seguinte mensagem “Nenhuma divisão com enfermeiros encontrada”.

3 - Reflexão Crítica

3.1 Introdução

Este capítulo surge com o objetivo de realizar uma avaliação crítica ao trabalho desenvolvido pelos membros do grupo. Em adição, confrontamos também os objetivos propostos versus os objetivos atingidos, e as dificuldades encontradas na conclusão dos mesmos.

3.2 Divisão do Trabalho

De modo a existir uma partição semelhante no que diz respeito ao desenvolvimento do projeto, optámos por dividir tarefas.

- Resolvido por Carlos Martins:
 - Resolução Pergunta 2: Em que tipo de sala estás agora?
 - Resolução Pergunta 6: Quanto tempo achas que falta até ficares sem bateria?
 - Resolução Pergunta 8: Se encontrases um enfermeiro numa divisão, qual é a probabilidade de estar lá um doente?
 -
- Resolvido por Rui Ferreira:
 - Resolução Pergunta 1: Qual foi a penúltima pessoa que viste?
 - Resolução Pergunta 7: Qual a probabilidade de encontrar um livro numa divisão, se já encontraste uma cadeira?

No entanto, apesar de ter existido esta divisão cooperamos sempre mutuamente entre ambos de modo a procurar soluções para os problemas que iam aparecendo com o evoluir do projeto.

3.3 Problemas Encontrados

Um dos principais problemas que verificámos, ao longo do desenvolvimento do projeto, consiste na classificação das várias salas existentes no hospital, na medida que o manuseamento do robot pelas respetivas salas este pode não reconhecer todos os objetos presentes, levando desta maneira a uma incorreta classificação. O mesmo acontece para o cálculo das probabilidades, uma vez que ao não serem reconhecido a totalidade de objetos existentes numa sala, pode induzir a erros de calculo.

3.4 Análise Crítica

Confrontado os objetivos propostos com os objetivos alcançados, concluímos que não atingimos o que inicialmente tínhamos idealizado. No entanto, estamos satisfeitos com o trabalho desenvolvido até ao momento.

Analisando criticamente os motivos que nos levaram a não completar as perguntas 3, 4 e 5 deve-se principalmente ao facto de não conseguirmos implementar grafos de forma correta. Idealizamos as respostas a estas perguntas usando grafos com pesos, em que o peso de cada aresta seria a distância euclidiana do centro da divisão em que o robô está ao centro da divisão anterior. O robô ao mudar de divisão adicionava uma nova aresta com essa informação. No caso da pergunta 3 usaríamos a função dijkstra para calcular o caminho ótimo, na pergunta 4 a soma do peso das arestas, com base na função dijkstra e na 5 a soma das arestas a dividir pelo tempo que o robô demora a fazer uma deslocação.

Relativamente à dinâmica do funcionamento do grupo, tentámos distribuir as tarefas de forma equivalente de modo a que ambos tivéssemos a mesma carga de trabalho, contudo estabelecíamos contacto regularmente com o intuito de cooperar na elaboração e resolução dos problemas que apareciam com o desenvolvimento do projeto. Resumidamente, podemos concluir que tivemos uma dinâmica bastante positiva.

4 - Conclusão

4.1 Conclusão

Apesar dos objetivos finais atingidos não serem os desejados, estamos satisfeitos com o trabalho elaborado e as aprendizagens adquiridas e solidificadas no desenvolvimento do projeto. Com a elaboração deste trabalho, conseguimos não só trabalhar como melhorar o conhecimento numa linguagem que nunca antes tinha sido abordada por nenhum membro do grupo, adquirindo uma experiência que pode vir a ser útil para futuros projetos.

No decorrer da realização do projeto, conseguimos adquirir uma perceção do funcionamento do mundo da inteligência artificial, concluindo que se para um simples projeto de um robot manuseado num ambiente controlado surgem algumas dificuldades na maneira de como implementar as funções para atingir o objetivo proposto, quando se tentar aplicar num mundo real e com um maior número de variáveis a ter em conta, maior o grau de exigência e dificuldade para construir funções que possam ser implementadas com sucesso.

4.2 Trabalho Futuro

Futuramente, um dos aspetos a melhorar no projeto desenvolvido assenta na resolução das questões que ficaram por resolver.

Outra dos aspetos que achamos também interessantes de explorar no futuro consiste na expansão das questões a responder pelo robot.

Referências

Luís Alexandre. Inteligência Artificial. Lecture Notes, 2020.

<https://www.w3schools.com/python/>