Universidade da Beira Interior

Departamento de Informática

Nº 1 - 2019: *Projeto Prático - Inteligência Artificial*

Elaborado por:

Diogo Dias, Nº35394 Pedro Marreiros, Nº36642

Orientador:

Professor Doutor Luís Alexandre

27 de Dezembro de 2019

Capítulo 1

Introdução

1.1 Enquadramento

Este trabalho, e respetivos anexos foram elaborados no âmbito da Unidade Curricular Inteligência Artificial, leccionada pelo Professor Doutor Luís Alexandre. Serve este trabalho para aprofundar os conhecimentos adquiridos no decorrer das aulas de uma maneira prática, permitindo aos alunos manipular o seu próprio robô.

1.2 Motivação

Nos últimos anos, a área dedicada ao estudo das Inteligências Artificiais deu um salto enorme. Hoje em dia, temos agentes capazes de derrotar humanos em jogos tais como xadrez, damas, DOTA2, Temos também agentes com utilizações práticas no nosso dia-a-dia, tais como a Alexa, feita pela Amazon ou até mesmo os robôs das linhas de produção. Assim, torna-se extremamente importante aprender esta área, dado o seu crescimento, sendo essa a principal motivação deste projeto.

1.3 Organização do Documento

De modo a refletir o trabalho que foi feito, este documento encontra-se estruturado da seguinte forma:

1. O primeiro capítulo – **Introdução** – apresenta o projeto, a motivação para a sua escolha, o enquadramento para o mesmo, os seus objetivos e a respetiva organização do documento.

2. O segundo capítulo –**Altrerações** – Aqui os autores irão descrever algumas alterações feitas à versão base do projeto, de modo a conseguir responder

às questões propostas.

3. O terceiro capítulo –**Questões** – Neste capítulo, os autores demonstram como foram respondidas as questões propostas pelo professor.

4. o quarto capítulo –**Conclusão** – Por fim, serve este capítulo para inferir sobre o trabalho feito, e concluir este documento.

Capítulo 2

Alterações

2.1 Introdução

Neste capítulo, são apresentadas algumas alterações feitas à base do projeto de modo a conseguir responder às perguntas

2.2 Lista de Divisões e Objetos

Dado que as paredes não são alteradas nos *datasets* de teste, os autores decidiram criar uma matriz onde cada linha representa uma divisão. Esta matriz contém o espaço das coordenadas de cada divisão, o seu tipo (quarto simples, quarto duplo, suite, etc), se já foi visitada anteriormente e a sua divisão anterior.

O trecho de código seguinte mostra a Lista de Divisões:

```
Divisao = [[-15.6,3.6,-3.1,-1.5,-1,0,-1],
    [-11.8,-9.6,-1.3,5.1,-1,0,-1],
    [-11.8,3.6,5.2,7.3,-1,0,-1],
    [-4,-1.4,-1.3,5.1,-1,0,-1],
    [-15.6,-12.4,-0.8,2.3,-1,0,-1],
    [-15.6,-12.4,2.9,7.3,-1,0,-1],
    [-15.6,-11.1,7.9,11.1,-1,0,-1],
    [-10.6,-6.2,7.9,11.1,-1,0,-1],
    [-5.5,-1.2,7.9,11.1,-1,0,-1],
    [-0.6,3.6,7.9,11.1,-1,0,-1],
    [-0.8,3.6,2.3,4.9,-1,0,-1],
    [-0.8,3.6,-0.8,1.7,-1,0,-1],
    [-9,-7,-0.8,4.9,-1,0,-1],
    [-6.5,-4.5,-0.8,4.9,-1,0,-1]]
```

Excerto de Código 2.1: Lista de Divisões.

Os autores decidiram também associar a cada divisão a sua respetiva lista de objetos. Assim, em qualquer momento da execução, o robô consegue saber que objetos estão nas divisões já visitadas do hotel. Ao visitar uma divisão, o robô irá adicionar os objetos presentas na mesma para a sua respetiva lista de objetos.

O trecho de código seguinte mostra a função addobj(id,nome):

```
def addobj(id,nome):
    for i in range(len(objecto[id])):
        if nome == objecto[id][i]:
            return
    objecto[id].append(nome)
    print objecto[id]
```

Excerto de Código 2.2: Função addobj(id,nome)

2.3 Reconhecimento de Divisões

Foi feita uma função que é executada conforme o robô se move. Esta função reconhece a divisão onde o robô está, e conforme os objetos presentes na sua respetiva lista, determina em que tipo de divisão está.

O trecho de código seguinte mostra a função divisao(x,y):

```
def divisao(x,y):
    global div_ant, div_ant_ant
    for i in range(14):
        if x >= Divisao[i][0] and x <= Divisao[i][1] and y >=
            Divisao[i][2] and y <= Divisao[i][3]:
             if i != div_ant:
        if Divisao[i][5] == 0:
           Divisao[i][5] = 1
           Divisao[i][6] = div_ant
                div_ant_ant = div_ant
                div_ant = i
             print "div %i di_ant %i" % (div_ant, div_ant_ant)
             if i == 0 or i == 1 or i == 2 or i == 3:
                Divisao[i][4] = 5
                return i
            camas = 0
            mesas = 0
            cadeiras = 0
            for j in range(len(objecto[i])):
                if "bed" in objecto[i][j]:
                    camas += 1
                elif "table" in objecto[i][j]:
                    mesas += 1
                elif "chair" in objecto[i][j]:
```

```
                cadeiras += 1
            if  Divisao[i][4] == 3:
                return i
            elif camas == 1:
                Divisao[i][4] = 1
            elif camas == 2:
                Divisao[i][4] = 2
            elif mesas == 1 and cadeiras > 1:
                Divisao[i][4] = 4
            else:
                Divisao[i][4] = 0
            if Divisao[div_ant_ant][4] != 5 and (Divisao[
               div_ant_ant][4] == 1 or Divisao[div_ant_ant][4]
               == 2 or Divisao[i][4] == 1 or Divisao[i][4] == 2)
               :
                print "Suites: %i%i" % (i, div_ant_ant)
                Divisao[i][4] = 3            #Suite
                Divisao[div_ant_ant][4] = 3
            return i
    return -1
```

Excerto de Código 2.3: Função divisao(x,y)

2.4 Conclusões

Com estas alterações feitas, os autores passaram então a responder às perguntas propostas pelo professor no enunciado do projeto.

Capítulo 3

Questões

3.1 Introdução

Neste capítulo, os autores irão descrever como foram respondidas as questões propostas no enunciado.

3.2 Pergunta 1 - Quantos quartos não estão ocupados?

Para responder a esta pergunta, é percorrida a lista de divisões e verifica-se se a divisão é um quarto. Se a divisão fôr um quarto, o total de quartos é incrementado, e é percorrida a sua lista de objetos e verifica-se se contém uma pessoa. Se sim, então o número de quartos ocupados é incrementado. No final, o cálculo dos quartos não ocupados é feito subtraindo o número de quartos ocupados ao número total de quartos. Esta pergunta foi feita pelo Diogo Dias.

O trecho de código seguinte mostra o código usado para responder à Pergunta 1:

```
    if int(data.data) == 1:
        cont_total = 0
        cont_pessoas = 0
        for i in range(3,14):
            if Divisao[i][4] != -1:
                cont_total += 1
            for j in range(len(objecto[i])):
                if "person" in objecto[i][j]:
                    cont_pessoas += 1
                    break
```

```
        print "Ate ao momento foram encontradas %i Divis es nao
            ocupadas." % (cont_total - cont_pessoas)
```

Excerto de Código 3.1: Pergunta 1 (Feita pelo Diogo Dias

3.3 Pergunta 2 - Quantas suites foram encontradas até agora?

Aqui, recorre-se à lista de divisões e verifica-se se a sua classificação é correspondente a uma suite. Esta pergunta foi feita pelo Diogo Dias.

```
    if int(data.data) == 2:
        cont_suites = 0
        for i in range (3,14):
            if Divisao[i][4] == 3:
                cont_suites += 1
        print "Ate ao momento foram encontradas %i Suites." % (
            cont_suites / 2)
```

Excerto de Código 3.2: Pergunta 2 (Feita pelo Diogo Dias

3.4 Pergunta 3 - É mais provável encontrar pessoas nos quartos ou nos corredores?

Para responder a esta pergunta, são verificadas as primeiras 4 divisões, que correspondem aos corredores, e é contabilizado o número de pessoas presentes. Após isso, são corridas as restantes divisões e contabilizado o número de pessoas presentes. Por fim, são calculadas as probabilidades e a resposta é dada consoante as mesmas. Esta pergunta foi feita pelo Pedro Marreiros.

```
    if int(data.data) == 3:
    cont_people_corridor = 0
    cont_people_room = 0
    cont_people_total = 0
    prob_room = 0
    prob_corridor = 0
        for i in range (3):
        for j in range (len(objecto[i])):
        if "person" in objecto[i][j]:
```

```
            cont_people_corridor += 1
    for i in range (4, 14):
        for j in range (len(objecto[i])):
        if "person" in objecto[i][j]:
            cont_people_room += 1
    cont_people_total = cont_people_corridor + cont_people_room
    if cont_people_total == 0:
       print "Ainda n o foram encontradas pessoas"
    else:
       prob_corridor = cont_people_corridor / cont_people_total
       prob_room = cont_people_room / cont_people_total
       if prob_corridor > prob_room:
          print "   mais prov vel encontrar pessoas nos
             corredores"
       else:
          print "   mais prov vel encontrar pessoas nos quartos
             "
    #PERGUNTA 6
    if int(data.data) == 6:
        current_div = divisao(x_ant,y_ant)
    if current_div == 0:
       print "J   se encontra na divis o do elevador"
    else:
       while current_div != 0:
          aux_div = current_div
              current_div = Divisao[aux_div][6]
          print "Divis o %i -> Divis o %i" % ((aux_div + 1), (
             current_div + 1))
```

Excerto de Código 3.3: Pergunta 3 (Feita pelo Pedro Marreiros

3.5 Pergunta 4 - Se queremos encontrar um computador, para que quarto vamos?

Para responder a esta pergunta, é feita uma passagem pela lista de divisões e pela lista de objetos e é são registados quantos computadores existem em cada tipo de divisão. Após isso, os tipos de divisão são comparados entre si e é decidido qual o tipo mais provável de conter um computador. Esta pergunta foi feita pelo Diogo Dias.

```
    if int(data.data) == 6:
        current_div = divisao(x_ant,y_ant)
    if current_div == 0:
       print "J   se encontra na divis o do elevador"
```

```
    else:
        while current_div != 0:
            aux_div = current_div
                current_div = Divisao[aux_div][6]
            print "Divis o %i -> Divis o %i" % ((aux_div + 1), (
                current_div + 1))
```

Excerto de Código 3.4: Pergunta 4 (Feita pelo Diogo Dias

3.6 Pergunta 5 - Qual o número do quarto simples mais próximo?

Esta pergunta foi feita pelo Diogo Dias.

```
if int(data.data) == 5:
        MIN = 9999
        MinID = 0
        for i in range(3,14):
            if Divisao[i][4] == 1 and i != divisao(x_ant, y_ant)
                :
                x = (Divisao[i][0] + Divisao[i][1]) / 2
                y = (Divisao[i][2] + Divisao[i][3]) / 2
                dist = math.sqrt(((x_ant - x)**2) + ((y_ant - y)
                    **2))
                #print dist
                if dist < MIN:
                    MIN = dist
                    MinID = i + 1
        if MinID > 0:
            print "O n mero do Quarto Simples mais pr ximo    ,
                 at   agora, %i." % MinID
        else:
            print "Ainda n o foram encontrados Quartos Simples.
                "
```

Excerto de Código 3.5: Pergunta 5 (Feita pelo Diogo Dias

3.7 Pergunta 6 - Como ir desde a divisão atual até ao elevador?

Nesta pergunta, começamos por determinar qual a divisão onde o robô se encontra. Depois, é verificada a sua divisão anterior, e assim sucessivamente até chegar

à divisão 1, que corresponde à divisão do elevador. Esta pergunta foi feita pelo Pedro Marreiros.

```
if int(data.data) == 6:
        current_div = divisao(x_ant,y_ant)
    if current_div == 0:
        print "J  se encontra na divis o do elevador"
    else:
        while current_div != 0:
            aux_div = current_div
                current_div = Divisao[aux_div][6]
            print "Divis o %i -> Divis o %i" % ((aux_div + 1), (
                current_div + 1))
```

Excerto de Código 3.6: Pergunta 6 (Feita pelo Pedro Marreiros

3.8 Pergunta 7 - Quantos livros é estimado encontrar nos próximos 2 minutos?

De modo a conseguir responder a esta pergunta, no ínicio da execução é guardado o momento de começo. Quando esta pergunta é chamada, é calculado o momento atual, e é verificado quantos livros foram encontrados entre o ínicio da execução até ao momento atual. Após isso, é calculada a estimativa. Esta pergunta foi feita pelo Pedro Marreiros.

```
if int(data.data) == 7:
        current_time = time.time() - start_time
    print "Time since start = %i" % current_time
    books_found_total = 0
    estimated_books = 0

    for i in range (14):
        for j in range(len(objecto[i])):
            if "book" in objecto[i][j]:
              books_found_total += 1

    estimated_books = (120 * books_found_total) / current_time

    print "Estima-se encontrar %f livros nos pr ximos 2 minutos
        " % estimated_books
```

Excerto de Código 3.7: Pergunta 7 (Feita pelo Pedro Marreiros

3.9 Pergunta 8 - Qual a probabilidade de encontrar uma mesa num quarto sem livros, mas com pelo menos uma cadeira?

Primeiramente, é contado o número total de objetos presentes em memória. Após isso é contado o número de livros. Com isto feito,é calculada a probababilidade de encontrar um livro. A probabilidade anterior irá servir para saber a probabilidade de não encontrar um único livro. De seguida é calculada a probabilidade de encontrar uma cadeira e a probabilidade de encontrar uma mesa. Posto isto, recorre-se a uma das fórmulas dadas nos apontamentos teóricos para responder à questão. Esta pergunta foi feita pelo Pedro Marreiros.

```
if int(data.data) == 8:
    prob_find_book = 0
    prob_not_find_book = 0
    prob_find_chair = 0
    prob_find_table = 0
    cont_books = 0
    cont_chair = 0
    cont_table = 0
    cont_total = 0

    for i in range(14):
       cont_total += len(objecto[i])

    for i in range(14):
       for j in range(len(objecto[i])):
          if "book" in objecto[i][j]:
           cont_books += 1

    prob_find_book = float(cont_books) / float(cont_total)
    prob_not_find_book = 1.0 - prob_find_book

    for i in range(14):
       for j in range(len(objecto[i])):
          if "chair" in objecto[i][j]:
           cont_chair += 1

    prob_find_chair = float(cont_chair) / float(cont_total)


    for i in range(14):
       for j in range(len(objecto[i])):
          if "table" in objecto[i][j]:
```

```
        cont_table += 1

    prob_find_table = float(cont_table) / float(cont_total)

    prob_result = (float(prob_find_table) * float(
        prob_find_chair) * float(prob_not_find_book)) / float(
        prob_not_find_book)

    print "A probabilidade     %f" % prob_result
```

Excerto de Código 3.8: Pergunta 8 (Feita pelo Pedro Marreiros

3.10 Conclusões

Juntando o conteúdo deste capítulo com o contéudo do capítulo anterior, o leitor já consegue perceber o raciocínio dos autores no que toca às suas respostas.

Capítulo 4

Conclusões e Trabalho Futuro

4.1 Conclusões Principais

Conclui-se este documento constatando o facto de que todas as perguntas propostas foram respondidas pela versão final do agente, recorrendo ao conhecimento adquirido no decorrer do período lectivo da Unidade Curricular.