

Projeto de COMPUTAÇÃO E PROGRAMAÇÃO

Documentação técnica

GRUPO Nº 26

100163

Duarte Moraes
duarte.morais@tecnico.ulisboa.pt

100198

Joana Caramelo
joana.caramelo@tecnico.ulisboa.pt

Mestrado Integrado em Engenharia Mecânica

Lisboa, dezembro de 2020

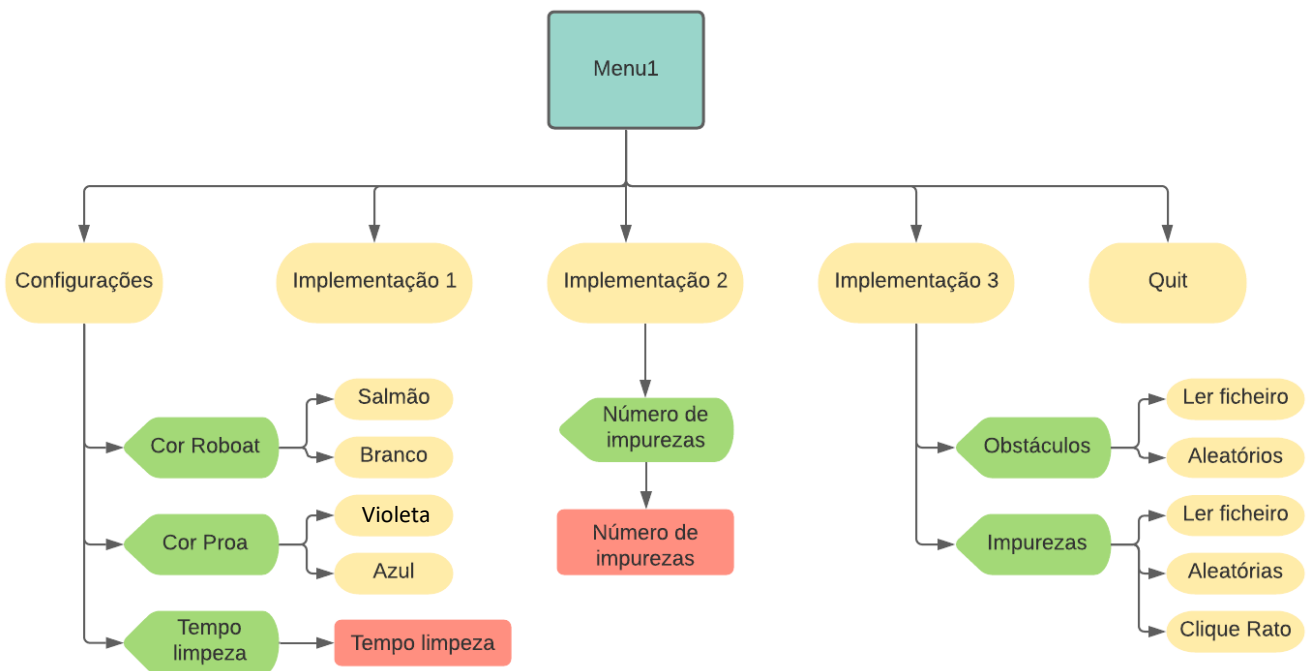
Índice

1. Documentação técnica	3
1.1. Arquitetura do programa.....	3
1.2. Projeto detalhado.....	6

1. Documentação técnica

1.1. Arquitetura do programa

Funcionamento do Menu Inicial



A amarelo – Botões

A verde – Opções de escolha

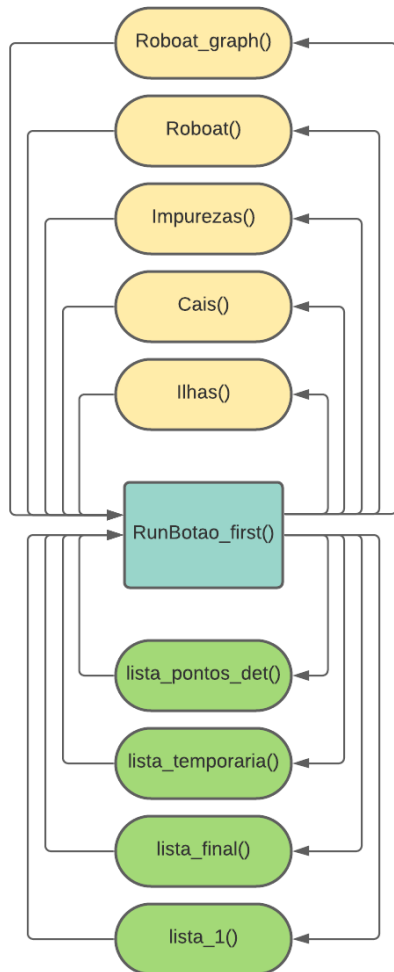
A vermelho - Entradas

O programa corre à base de escolhas feitas pelo utilizador, sendo que abre determinadas janelas à medida que o utilizador clica em certos botões.

São utilizadas funções **while** (de ciclo infinito) para que a função que corre o menu aguarde pelas escolhas feitas pelo utilizador.

Quando o utilizador clica num botão, gera então outro ciclo **while**. Isto pode aplicar-se a qualquer janela que for aberta.

Implementação 1 (RunBotao_first)

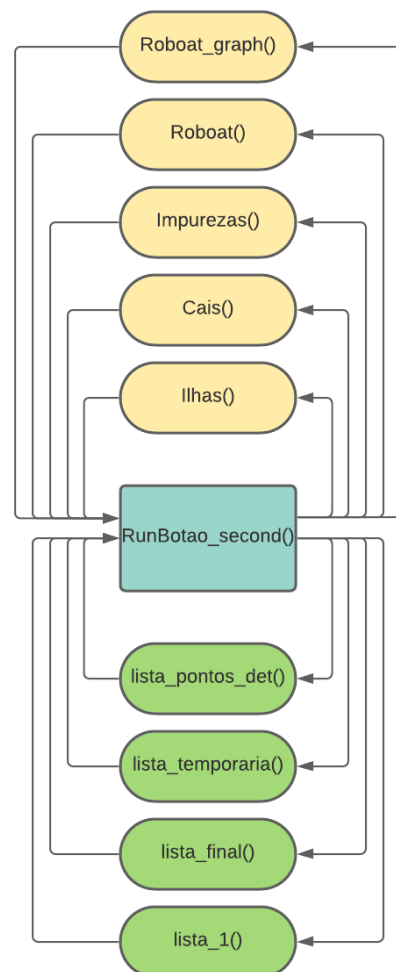


A amarelo – Métodos das classes “Ilhas”, “Impurezas”, “Cais”, “Roboat”.

A azul – Função que corre a primeira implementação.

A verde – Funções importadas do ficheiro “**Aux_Functions**”.

Implementação 2 (RunBotao_second)



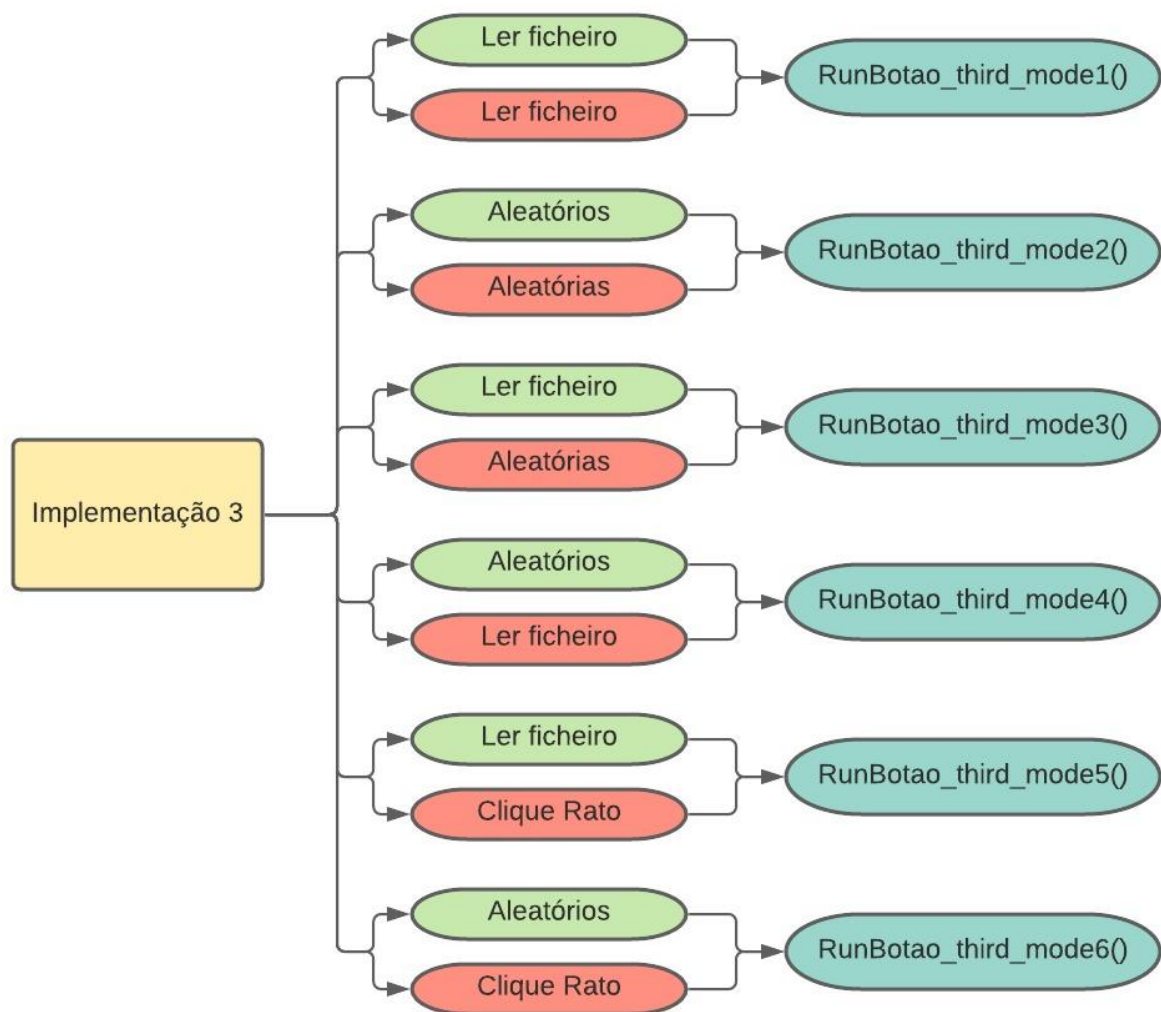
A amarelo – Métodos das classes “Ilhas”, “Impurezas”, “Cais”, “Roboat”.

A azul – Função que corre a segunda implementação.

A verde – Funções importadas do ficheiro “**Aux_Functions**”.

Da função “RunBotao_first” para a função “RunBotao_second” só varia a variável “nImpurezas”.

Implementação 3



A amarelo – botão escolha “Implementação 3”.

A verde – botões escolha obstáculos.

A vermelho – botões escolha impurezas.

A azul – funções que correm a terceira implementação em função das escolhas do utilizador.

1.2. Projeto detalhado

1.2.1 Módulo Botao

Tipo: Classe

Função: A classe botão desenha botões (retângulos legendados) na janela gráfica. Os botões permitem ao utilizador fazer diversas escolhas ao longo do programa.

a) Parâmetros: win, center, width, height, label, cor

b) Métodos:

 __init__ - inicializa a classe "Botao".

 clicked - reconhece quando se clica num botão.

 activate - ativa o botão.

 deactivate - desativa o botão.

 grablabel - permite legendar o botão.

1.2.2 Módulo Menu1

Tipo: Classe

Função: O Menu1 permite ao utilizador escolher qual das três implementações pretende visualizar.

a) Parâmetros: setSize

b) Métodos:

 __init__ - inicializa a classe "Menu1".

 use - deteta que botão é clicado, retornando "Implementacao 1", ou "Implementacao 2", "Menu 3", "Configurações" ou "Quit".

 close - fecha a janela.

1.2.3 Módulo Menu2

Tipo: Classe

Função: O Menu2 permite ao utilizador escolher o número de impurezas que deseja colocar no ambiente gráfico para que o Roboat as possa limpar.

a) Parâmetros: nImpurezas

b) Métodos:

 __init__ - inicializa a classe "Menu2".

 get_nImpurezas - método que retorna o número de impurezas.

 use - deteta que botão é clicado, retornando "Seguinte" ou "Quit".

 close - fecha a janela.

1.2.4 Módulo Menu3

Tipo: Classe

Função: O menu 3 permite escolher se os obstáculos da terceira implementação são lidos de um ficheiro ou gerados aleatoriamente e se as impurezas são lidas de um ficheiro, geradas aleatoriamente ou com cliques do rato.

a) Parâmetros: opcao_obstaculos , opcao_Imp

b) Métodos:

 __init__ - inicializa a classe "Menu3".

 get_opcao_Imp – método que retorna a opção escolhida pelo utilizador para as impurezas

 get_opcao_obstaculos – método que retorna a opção escolhida pelo utilizador para os obstáculos.

 use – deteta que botão é clicado, retornando "Seguinte" ou "Quit" e ativando/desativando botões para retornar as opções escolhidas.

 close – fecha a janela.

1.2.5 Módulo Configuracoes

Tipo: Classe

Função: A classe configurações obtém valores, tais como a cor do roboat, a cor da proa, o número de impurezas existentes e o tempo que o roboat demora a limpar as impurezas.

a) Parâmetros: cor_roboat, cor_proa, tempolimpeza

b) Métodos:

__init__ - inicializa a classe "Configuracoes".
get_tempo – retorna o tempo de limpeza.
get_cor_proa – retorna a cor da proa.
get_cor_roboat – retorna a cor do roboat.
use – Método que deteta que botão é clicado, retornando "Seguinte" ou "Quit" e ativando/desativando botões para retornar as cores escolhidas.
close – fecha a janela.

1.2.6 Módulo Impureza

Tipo: Classe

Função: Gerar os detritos em ambiente gráfico.

a) Parâmetros: centro2 , fill , outline , win

b) Métodos:

__init__ - inicializa a classe "Impureza".
desenhar_impurezas - permite escolher qual é o formato da ilha que será desenhada
getCentro - retorna o centro da impureza
undraw_imp – apaga a impureza graficamente

1.2.7 Módulo Ilhas

Tipo: Classe

Função: Gerar as ilhas (obstáculos) das quais o Roboat terá que se desviar para limpar os detritos.

a) Parâmetros: centro , fill , outline , win , opcao

b) Métodos:

__init__ - inicializa a classe "Ilhas".
desenhar_ilha – permite escolher qual o formato da ilha que será desenhada.
quadrado_ilha_graph – desenha uma ilha com formato quadrangular.
circulo_ilha_graph – desenha uma ilha com formato circular.
triangulo_ilha_graph – desenha uma ilha com formato triangular.
cruz_ilha_graph – desenha uma ilha com o formato de uma cruz.

1.2.8 Módulo Cais

Tipo: Classe

Função: Gerar o cais do Roboat através de retângulos.

a) Parâmetros: point3 , point4 , fill2 , outline2 , win

b) Métodos:

__init__ - inicializa a classe "Cais".
desenhar_cais - desenha graficamente o cais através de retângulos.

1.2.9 Módulo Roboat

Tipo: Classe

Função: Classe que cria as propriedades do Roboat

a) Parâmetros: tempo , lista_i , destino , angulo3

b) Métodos:

__init__ - inicializa a classe "Roboat".
getInicio –retorna a posição inicial e o angulo inicial do Roboat
velocidade – define a velocidade do Roboat para todas as circunstâncias que possam acontecer na plataforma gráfica.
mover – calcula o vetor velocidade do Roboat em linha reta para o destino
angulo_2 – define o ângulo do vetor velocidade, considerando o centro do Roboat como referencial.
setAngulo – permite redefinir o angulo do vetor velocidade.
update – atualiza a posição e o vetor velocidade do Roboat a cada instante.
getY – retorna o valor da ordenada do centro do Roboat.
getX – retorna o valor da abcissa do centro do Roboat.
getAngulo – retorna o valor do ângulo do vetor velocidade.

Explicação do movimento adotado pelo Roboat quando encontra um obstáculo:

A velocidade que o Roboat adota quando se depara com um obstáculo é calculada pelas seguintes expressões:

$$x = \arctan\left(\frac{y_r - y_i}{x_r - x_i}\right)$$

sendo y_r : ordenada do centro do Roboat
 y_i : ordenada do centro da ilha
 x_r : abscissa do centro do Roboat
 x_i : abscissa do centro da ilha

Pelo que, se o ponto C (da figura à direita) representar o centro da ilha e o ponto B representar o centro do Roboat, então $\frac{y_r - y_i}{x_r - x_i}$ é o declive da reta CB.

Tendo o ângulo x , é possível calcular, assim, o ângulo alfa (α) da seguinte maneira:

$$\alpha = \arctan\left(\frac{\operatorname{cosec}(x)}{\sec(x)}\right)$$

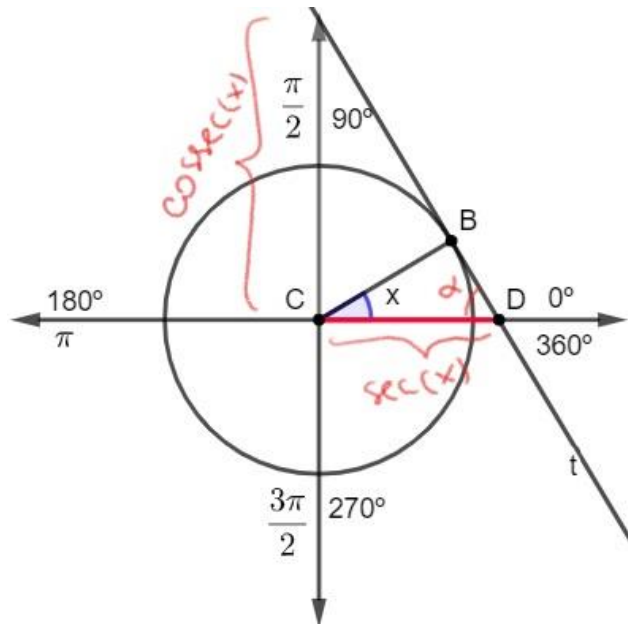
$$\cos(\alpha) \cdot v = vx$$

$$\sin(\alpha) \cdot v = vy$$

O facto do valor alfa mudar ao longo do tempo (à medida que o Roboat contorna o obstáculo), implica que o vetor velocidade também mude ao longo do tempo. O vetor velocidade é sempre um vetor tangente à circunferência imaginária que delimita o obstáculo.

É importante reter que existem dois casos diferentes para o cálculo deste alfa, pois quando o alfa se encontra no Segundo e terceiro quadrante tem de se adicionar pi ao valor do ângulo x .

No caso de a abscissa do roboat ser igual à abscissa do centro da ilha o valor de alfa é zero pois o valor de x não está definido já que $\arctan\left(\frac{\pi}{2}\right) = +\infty$ e $\arctan\left(-\frac{\pi}{2}\right) = -\infty$.



```
if ((a - self.xpos)**(2) + (b - self.ypos)**(2))**1/2 <= 49 and self.xpos > a :  
  
    # (... declive)  
    x1 = atan((b-self.ypos)/(a-self.xpos))  
  
    alfa = atan((1/cos(x1))*(sin(x1)))  
  
    vel = 40  
    self.xvel=vel*cos(alfa)  
    self.yvel= vel*sin(alfa)  
    valor = True
```

- Situação para quando o centro do Roboat tem maior abscissa do que o centro da ilha (primeiro e quarto quadrantes).


```
elif ((a - self.xpos)**(2) + (b - self.ypos)**(2))**1/2 <= 49 and self.xpos < a:

    x1 = atan((b-self.ypos)/(a-self.xpos))
    alfa = atan((1/cos(x1+pi))*(sin(x1+pi)))

    vel = 40
    self.xvel=-vel*cos(alfa)
    self.yvel=-vel*sin(alfa)
    valor = True
```

- Situação para quando o centro do Roboat tem menor abcissa do que o centro da ilha (segundo e terceiro quadrantes).

```
elif ((a - self.xpos)**(2) + (b - self.ypos)**(2))**1/2 <= 49 and self.xpos == a:

    alfa = 0

    vel = 40
    self.xvel=vel*cos(alfa)
    self.yvel= vel*sin(alfa)
    valor = True
```

- Situação para quando o centro do Roboat tem abcissa igual ao centro da ilha (quando x não está definido).

1.2.10 Módulo Roboat_graph

Tipo: Classe

Função: Classe responsável por criar gráficamente o Roboat

a) Parâmetros: self , raio1 , centro1 , cor_roboat , cor_proa , outline3 , tempo , win , listai . pos , dt

b) Métodos:

__init__ - inicializa a classe "Roboat_graph".

desenhar – método que desenha o Roboat e a proa do Roboat.

mover_roboat – o roboat irá sofrer uma atualização da sua posição ao longo do tempo.

triangulo_graph – elabora e desenha uma proa triangular, de maneira a que aponte sempre para a mesma direção do vetor velocidade ao longo do tempo.

bola_graph – desenha o "corpo" circular do Roboat em ambiente gráfico.

velocidade_graph – método que "chama" o método velocidade da Class Roboat para que o Roboat tenha velocidade em ambiente gráfico.

mover_fim_graph – método que faz com que o Roboat se mova até ao Cais após a limpeza das impurezas.

update – atualiza constantemente o local do Roboat em ambiente gráfico.

getX – retorna a abcissa do Roboat em ambiente gráfico.

getY – retorna a ordenada do Roboat em ambiente gráfico.

undraw – apaga em ambiente gráfico a proa e o Roboat.

1.2.11 Módulo RunBotao_first

Tipo: Função

Função: Função que irá correr a primeira implementação.

Classes importadas:

- Classe "Ilhas"
- Classe "Cais"
- Classe "Impurezas"
- Classe "Roboat"
- Classe "Roboat_graph"

Outras funções importadas: (do ficheiro Aux_Functions)

- lista_pontos_det_imp1
- lista_temporaria
- lista_final
- lista_1

```

window = GraphWin("Piscina do Dudu e da Juju",800,800,autoflush=False)
window.setCoords(0,0,100,100)
window.setBackground("Darkturquoise")

```

- Criação da janela gráfica que representa uma piscina (ambiente gráfico onde o Roboat efetua a sua função).

```

window.getMouse()
lista_pontos_ilhas = [Point(25,75),Point(25,25),Point(75,75),Point(75,25),Point(50,50)]
lista_opcao_ilhas = ["Triangulo","Circulo","Circulo","Quadrado","Cruz"]

islands_list = lista_1(window, lista_pontos_ilhas, lista_opcao_ilhas)

```

- Com um clique do rato as ilhas aparecem no ambiente gráfico.
- Elaboração da lista com os centros das ilhas e da lista com os formatos das ilhas. A partir dessas duas listas a função lista_1 irá gerar as ilhas em ambiente gráfico.

```

window.getMouse()
Cais1 = Cais(Point(0,0),Point(15,5) , 'black', 'white', window)
Cais2 = Cais(Point(0,5),Point(5,15) , 'black', 'white', window)

```

- Com um clique do rato o Cais aparece no ambiente gráfico.
- As diferentes partes do Cais (dois retângulos) são criadas a partir da Classe "Cais".

```

Roboat1 = Roboat_graph(3,Point(40,50), cor_roboat, cor_proa , "black",5,window,islands_list)

```

- Criação do Roboat em ambiente gráfico a partir da classe "Roboat_graph".

```

a = lista_pontos_det(lista_pontos_ilhas,window)
b = lista_temporaria(Roboat1,a)
c = lista_final(b)

```

- Criação das impurezas e das ilhas em ambiente gráfico. (É possível ver a descrição destas funções nos módulos lista_pontos_det, lista_temporária, lista_final_ lista_1).

```

lista_detritos = []
for d in c:

    pos1 = Point(d.getX(), d.getY())
    i1 = Impureza(pos1, 'white', 'black', window)
    lista_detritos.append(i1)

```

- A partir da lista com os centros das impurezas ordenados, irá ser criada uma lista com Impurezas, feitas a partir da classe Impureza.

```

for x in lista_detritos:

    Roboat1.mover_roboat(x.getCentro())
    sleep(tempo_limpeza)
    x.undraw_imp()

```

- O Roboat irá mover-se até ao centro de cada impureza e demorar um certo tempo a limpar cada impureza.

```
Roboat1.mover_fim_graph()  
window.getMouse()
```

- O Roboat regressa ao Cais após limpar todas as impurezas.

1.2.12 Módulo lista_pontos_det:

Tipo: Função

Função: Escolhe, através de um clique do rato, o lugar das impurezas em ambiente gráfico. Esta função cria restrições para que não seja possível criar impurezas em cima de ilhas, do Cais e da zona de partida do Roboat, e ainda uma restrição que impede o Roboat saia da janela gráfica (isto é, restrição que impede a criação de impurezas a uma certa distância da extremidade da janela gráfica).

Entradas: lista_pontos_ilhas , window , ponto_d

Saídas: lista_pontos_detritos

1.2.13 Módulo lista_pontos_det_imp1:

Tipo: Função

Função: Escolhe, através de um clique do rato, o lugar de uma impureza em ambiente gráfico. Esta função cria restrições para que não seja possível criar impurezas em cima de ilhas, do Cais e da zona de partida do Roboat, e ainda uma restrição que impede o Roboat saia da janela gráfica (isto é, restrição que impede a criação de impurezas a uma certa distância da extremidade da janela gráfica). Esta função só permitir criar uma impureza em ambiente gráfico.

Entradas: lista_pontos_ilhas , window , ponto_d

Saídas: lista_pontos_detritos

1.2.14 Módulo lista_pontos_det_imp2:

Tipo: Função

Função: Escolhe, através de um clique do rato, o lugar das impurezas em ambiente gráfico. Esta função cria restrições para que não seja possível criar impurezas em cima de ilhas, do Cais e da zona de partida do Roboat, e ainda uma restrição que impede o Roboat saia da janela gráfica (isto é, restrição que impede a criação de impurezas a uma certa distância da extremidade da janela gráfica). A diferença entre esta função e a função “lista_pontos_det_imp1” consta no facto de esta função permitir criar diversas impurezas em ambiente gráfico.

Entradas: lista_pontos_ilhas , window , ponto_d, nImpurezas

Saídas: lista_pontos_detritos

1.2.15 Módulo lista_temporaria:

Tipo: Função

Função: Elaboração de uma lista temporária com pares ordenados dos centros das impurezas e as suas distâncias em relação à posição inicial do roboat. Ordenação da lista temporária de maneira a que os pontos fiquem ordenados pela distância que têm em relação ao ponto inicial

Entradas: Roboat1, lista_pontos_detritos

Saídas: lista_pontos_temporaria

1.2.16 Módulo lista_final:

Tipo: Função

Função: Eliminação das distâncias na lista temporária de modo a que se obtenha uma lista apenas com os centros das impurezas ordenadas.

Entradas: lista_pontos_temporaria

Saídas: lista_final

1.2.17 Módulo lista_1:

Tipo: Função

Função: Gerar todas as ilhas em ambiente gráfico

Entradas: win , lista_pontos1_ilhas , lista_opcao1_ilhas

Saídas: lista_ilhas

1.2.18 Módulo RunBotao_second:

Tipo: Função

Função: Função que irá correr a segunda implementação.

Classes importadas:

- Classe "Ilhas"
- Classe "Cais"
- Classe "Impurezas"
- Classe "Roboat"
- Classe "Roboat_graph"

Outras funções importadas: (do ficheiro Aux_Functions)

- lista_pontos_det_imp2
- lista_temporaria
- lista_final
- lista_1

A função referente à segunda implementação é em quase tudo semelhante à primeira, apenas variando na função que define o número de impurezas limpas pelo Roboat:

```
def lista_pontos_det_imp2(lista_pontos_ilhas,window,nImpurezas):
```

A função "lista_pontos_det_imp2" (importada do ficheiro Aux_Functions) tem mais uma variável comparativamente à primeira implementação: a variável nImpurezas (número de impurezas).

1.2.19 Módulo RunBotao_third_mode1:

Tipo: Função

Função: Função que irá correr a terceira implementação quando tanto os obstáculos como as impurezas são lidos de um ficheiro.

Classes importadas:

- Classe "Ilhas"
- Classe "Cais"
- Classe "Impurezas"
- Classe "Roboat"
- Classe "Roboat_graph"

Outras funções importadas

- Ficheiro Aux_Functions

1.2.20 Módulo RunBotao_third_mode2:

Tipo: Função

Função: Função que irá correr a terceira implementação quando tanto os obstáculos como as impurezas são lidos gerados aleatoriamente.

Classes importadas:

- Classe "Ilhas"
- Classe "Cais"
- Classe "Impurezas"
- Classe "Roboat"
- Classe "Roboat_graph"

Outros ficheiros importados:

- Ficheiro Aux_Functions

- Functions_ficheiros_aleatorios

1.2.21 Módulo RunBotao_third_mode3:

Tipo: Função

Função: Função que irá correr a terceira implementação quando os obstáculos são lidos de um ficheiro e as impurezas são criadas aleatoriamente.

Classes importadas:

- Classe "Ilhas"
- Classe "Cais"
- Classe "Impurezas"
- Classe "Roboat"
- Classe "Roboat_graph"

Outras funções importadas

- Ficheiro Aux_Functions
- Functions_ficheiros_aleatorios

1.2.22 Módulo RunBotao_third_mode4:

Tipo: Função

Função: Função que irá correr a terceira implementação quando os obstáculos são gerados aleatoriamente e as impurezas são lidas de um ficheiro.

Classes importadas:

- Classe "Ilhas"
- Classe "Cais"
- Classe "Impurezas"
- Classe "Roboat"
- Classe "Roboat_graph"

Outras funções importadas

- Ficheiro Aux_Functions
- Functions_ficheiros_aleatorios

1.2.23 Módulo RunBotao_third_mode5:

Tipo: Função

Função: Função que irá correr a terceira implementação quando os obstáculos são lidos de um ficheiro e as impurezas são geradas através de cliques do rato.

Classes importadas:

- Classe "Ilhas"
- Classe "Cais"
- Classe "Impurezas"
- Classe "Roboat"
- Classe "Roboat_graph"

Outras funções importadas

- Ficheiro Aux_Functions)
- Functions_ficheiros_aleatorios

1.2.24 Módulo RunBotao_third_mode6:

Tipo: Função

Função: Função que irá correr a terceira implementação quando os obstáculos são gerados aleatoriamente e as impurezas são criadas através de cliques do rato.

Classes importadas:

- Classe "Ilhas"

- Classe "Cais"
- Classe "Impurezas"
- Classe "Roboat"
- Classe "Roboat_graph"

Outros ficheiros importados:

- Ficheiro Aux_Functions
- Functions_ficheiros_aleatorios

1.2.25 Módulo main():

Tipo: Função

Função: Esta função tem o objetivo de correr o programa, assim esta abre o menu principal e corre em função das escolhas que o utilizador faz no mesmo.

Esta função importa todas as classes e funções do programa, caso contrário, o mesmo não funcionaria.

Ficheiro Functions_ficheiros_aleatorios:

Funções:

- impurezas_aleatorias(lista_pontos_ilhas,window)
impurezas são geradas aleatoriamente
- ilhas_aleatorias_pontos()
os centros das ilhas são gerados aleatoriamente
- ilhas_aleatorias_pontos_mode4()
os centros das ilhas são gerados aleatoriamente, sendo que não se podem gerar ilhas nos locais das impurezas definidas pelo ficheiro "Documento_leitura_impurezas.txt"
- ilhas_aleatorias_formato()
o formato das ilhas é gerado aleatoriamente
- leitura (Documento_leitura)
é feita a leitura do tamanho da janela gráfica através de um ficheiro txt
- leitura_opcao_centros(Documento_leitura)
é feita a leitura dos centros das ilhas e do seu formato através de um ficheiro txt
- leitura_cais(Documento_leitura)
é feita a leitura dos pontos do cais através de um ficheiro txt
- leitura_impurezas (Documento_leitura)
é feita a leitura das impurezas através de um ficheiro txt