

Programação avançada - UFPE 2022.2

Projeto 1: O robô Roomba

- Leia o documento inteiro antes de começar a trabalhar.
- Esboce sua solução antes de iniciar a programação.
- O trabalho é feito em duplas.
- Não há necessidade de apresentar um relatório.

Fundo do projeto

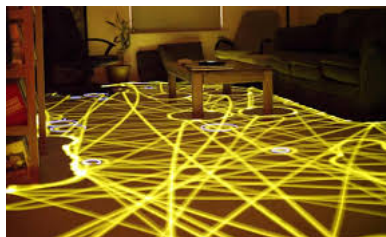
O objetivo deste projeto é desenvolver um simulador do robô Roomba 1(a). O robô Roomba é um robô aspirador que utiliza uma estação 1(b) como ponto de partida e carrega sua bateria. A partir desta estação, o robô se moverá aleatoriamente a fim de cobrir toda a superfície a ser limpa 1(c). Para isso, o robô escolhe uma direção aleatória, depois se move em linha reta até encontrar um obstáculo. Os obstáculos são detectados usando um sensor a bordo, como um pára-choque ou um laser. O robô limpa a superfície até que sua bateria atinja um nível crítico. Neste ponto, ele retorna à estação para carregar sua bateria. Se a bateria estiver completamente descarregada, o robô para.



(a) O robô Roomba



(b) A estação de carregamento



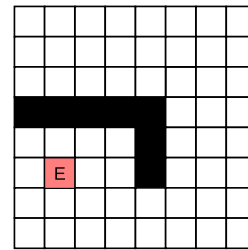
(c) O meio ambiente

Figura 1: Roomba

Descrição da arquitetura do código

Classe ambiente

- Membros da classe **ambiente**.
 - Atributos: dimensões da grade em x e y.
 - Atributos: grade representando o ambiente.
 - Atributos: posição da estação de carregamento.
 - Método: construtor usando dados armazenados em um arquivo.
 - Método: adicionar um obstáculo ao ambiente (1 célula).
 - Método: adicionar um obstáculo ao ambiente (retângulo).
 - Método: << imprimir o ambiente no terminal.



(a) O ambiente (célula branca: livre - célula preta: obstáculo - E: estação)

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(b) Grade representando o ambiente (0: célula livre - 1: obstáculo)

Figura 2: O ambiente e sua representação

Classe robô

- Membros da classe **robô**.
 - Atributos: nome.
 - Atributos: posição na grade em x e y.
 - Atributos: bateria.
 - Método: construtor usando dados armazenados em um arquivo.
 - Método: parar.
- O robô para quando a bateria é descarregada.

Classe modelo1

- Herdada da classe **robô**.
- Membros da classe **modelo1**.
 - Atributo: pára-choque.
 - Método: construtor.
 - Método: limpar.
- Este modelo pode se mover instantaneamente nas 4 direções mostradas na figura 3.

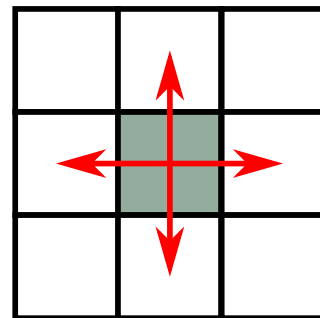
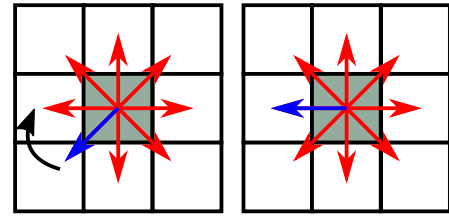


Figura 3: Possíveis deslocamentos para o modelo 1

Classe modelo2

- Herdada da classe **robô**.
- Membros da classe **modelo2**.
 - Atributo: orientação.
 - Atributo: laser.
 - Método: construtor.
 - Método: limpar.
- Este modelo pode se mover nas 8 direções mostradas nas figuras 4.
- Ele não pode se mover instantaneamente em qualquer direção. Tem que virar primeiro.



(a) Nesta posição o robô pode se mover de acordo com a flecha azul, ou girar
 (b) Nesta nova posição o robô pode se mover na direção da nova flecha azul, ou girar novamente

Figura 4: Possíveis deslocamentos para o modelo 2

Classe bateria

- Membros da classe **bateria**.
 - Atributo: nível.
 - Método: construtor.
 - Método: descarregar.
 - Método: carregar.
- A bateria é descarregada por uma unidade com cada ação.

Classe paracheque

- Membros da classe **paracheque**.
 - Método: calcular colisões.
- O pára-choque detecta a presença de um obstáculo na direção do movimento (Fig. 5).

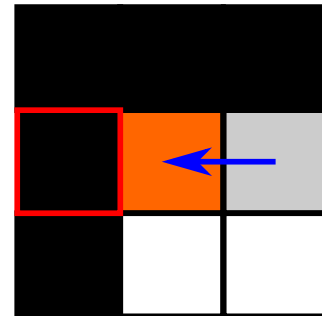


Figura 5: Quando o robô se move da célula cinza para a célula laranja, o pára-choque detecta o obstáculo em vermelho.

Classe laser

- Membros da classe **laser**.
 - Método: calcular colisões.
- O laser detecta a presença de um obstáculo para qualquer célula adjacente (Fig. 6).

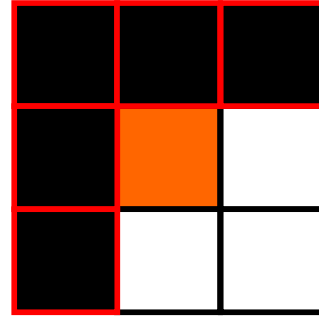


Figura 6: Quando o robô está na célula laranja, o laser detecta os obstáculos em vermelho.

Comentários

- As classes devem ser implementadas primeiro, depois é necessário criar uma ou mais funções **main** para testar as diferentes funções.
- Os membros variáveis não podem ser públicos.
- Esta arquitetura não é definitiva. Pode ser necessário acrescentar métodos que retornem valores de variáveis não públicas.
- Ao limpar, o robô escolhe sua direção de forma aleatória. No entanto, não pode escolher uma direção que leve a uma colisão.
- Teste o programa com o ambiente da figura (2). O programa deve funcionar com um ambiente diferente.