

Programação robótica

Residencia 2023

- **Leia o documento inteiro antes de começar a trabalhar.**
- **Esboce sua solução antes de iniciar a programação.**

Descrição do trabalho

O robô TIAGo está preso em um mini-labirinto. Para sair, tem que seguir estas regras:

- 1/ Quando parado, tem que tomar uma decisão:
 - Se houver uma parede em ambos os lados do robô (menos de 1,5 metros), ele deve se mover em linha reta;
 - Se houver uma parede de um lado e outra na frente, ela deve virar em direção ao espaço livre;
 - Se não houver parede em ambos os lados (menos de 1,5 metros), deve virar a cabeça em + e - 90 graus para tirar uma foto.
- 2/ A rotação é realizada por meio de um controlador baseado na odometria. O robô sempre gira em + ou - 90 graus.
- 3/ O deslocamento linear é realizado por meio de um controlador baseado na distância fornecida pelo laser. O robô para quando a distância é menor que um metro.
- 4/ Depois de tirar uma foto, o robô deve processá-la.
 - Se ele contém um pôster vermelho, o robô não pode virar naquela direção.
 - Se ele contiver um pôster verde, o robô deve virar nessa direção.
 - Se não houver pôster, o robô cumpriu a missão

Como organizar o código

- Você tem que implementar essas regras nos arquivos `main.py` e `camera.py`.
- Os dados de odometria e do laser são coletados por meio de dois `subscriber` no arquivo `main.py`
- Os dados da câmera são coletados por meio de um `subscriber` no arquivo `camera.py`
- As mensagens de controle da base móvel e da cabeça são enviados por meio de dois `publisher` no arquivo `main.py`
- Os processamentos em relação aos pontos 1, 2 e 3 são executados em três funções definidas no arquivo `main.py`
- O processamento de imagem é executado por meio de um serviço definido no arquivo `camera.py`
- A programação orientada ao objeto deve ser usada.

Thresholding

- No processamento de imagem digital, o thresholding é o método mais simples de segmentar imagens.
- Os métodos mais simples substituem cada pixel em uma imagem por um pixel preto se a intensidade da imagem $I_{i,j}$ é menor do que alguma constante fixa T (isto é, $I_{i,j} < T$), ou um pixel branco se a intensidade da imagem for maior do que essa constante.
- Para uma imagem colorida, um limite diferente pode ser usado para cada camada ou pode ser aplicado em uma única camada.
- O resultado final é uma imagem com uma camada única.

Instalação

- No workspace de TIAGo (nesse exemplo o nome do workspace é tiago_ws)
 - Copiar os arquivos **maze.world**, **mazeTest1.world** e **mazeTest2.world** em:
`tiago_ws/src/tiago_simulation/tiago_gazebo/worlds/`
 - Copiar os diretórios **maze**, **wall**, **landmark_green**, **landmark_red**, **frontWall** e **cornerWall** em:
`tiago_ws/src/tiago_simulation/tiago_gazebo/models/`
- Para iniciar a simulação, insira a seguinte linha (não se esqueça do comando no diretório do workspace):
`roslaunch tiago_gazebo tiago_gazebo.launch public_sim:=true robot:=titanium world:=maze`
- Existem ambientes simplificados para teste:
`roslaunch tiago_gazebo tiago_gazebo.launch public_sim:=true robot:=titanium world:=mazeTest1`
`roslaunch tiago_gazebo tiago_gazebo.launch public_sim:=true robot:=titanium world:=mazeTest2`
- Seu código deve fazer parte de um workspace diferente

Avaliação

- O trabalho é realizado em dupla.
- Os programas (pacotes) devem ser colocados no Classroom.
- Nenhum relatório será necessário.
- Os programas devem funcionar em um ambiente diferente respeitando as mesmas regras.