|  |  |
| --- | --- |
|  | UNIVERSIDADE ESTADUAL DE MARINGÁ  **Centro de Tecnologia - CTC**  **Departamento de Informática - DIN**  Programa de Pós-Graduação em Ciência da Computação - PCC |

RELATÓRIO

Disciplina: Robótica Inteligente e Autônoma

Código: DIN4091

Docente: Rodrigo Calvo

Discente: Leonardo S. S. da Rocha

R.A.: pg906715

**Conteúdo ministrado**

Tipos de robôs, sensores e atuadores

Controles reativo, deliberativo e hierárquico

Mapeamento de ambiente

Construção de mapas: métricos, topológicos e baseados em características

Localização de robôs

Algoritmos de navegação e baseados em grafos

Sistemas de múltiplos robôs

Projetos com simulador CoppeliaSim

**Desenvolvimento:**

O projeto consiste na realização de mapeamento e navegação com plotagem dos dados coletados.

O projeto está disponível em repositório no github pelo endereço:

< <https://github.com/leonardossrocha/RIA_UEM> >

**Arquivos:**

O projeto conta com arquivos do simulador CoppeliaSim e código utilizando a linguagem Python. A conexão entre simulador e IDE foi realizada via remoteApi do Coppelia que possibilita a conexão entre os dois softwares. A documentação do remoteApi está disponível em: < https://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsPython.htm >

Arquivos do RemoteAPI necessários no projeto:

remoteApi.so

sim.py

simConst.py

Descrição do código:

O código principal (avoidObstacle) segue comentado e conta com estrutura simples, fazendo uso dos módulos math, numpy, matplotlib e time.

De início, apresenta um método responsável para a leitura e guarda do sensor (fastHokuyo) do Pioneer. Além disso, guarda o ângulo dessa leitura.

Conta com métodos de:

leitura do sensor

Identificação de obstáculos com o laser do sensor

implementação da função de Bresenham.

Constante no código, a parte do script que é responsável por iniciar o programa, está estruturado num laço de condição simples onde todo o comportamento do código é iniciado ou, encerrado, de acordo com a condição estabelecida.

Demais detalhes estão disponíveis nos comentários inseridos no código e no arquivo README do repositório de armazenamento do código criado no GitHub.