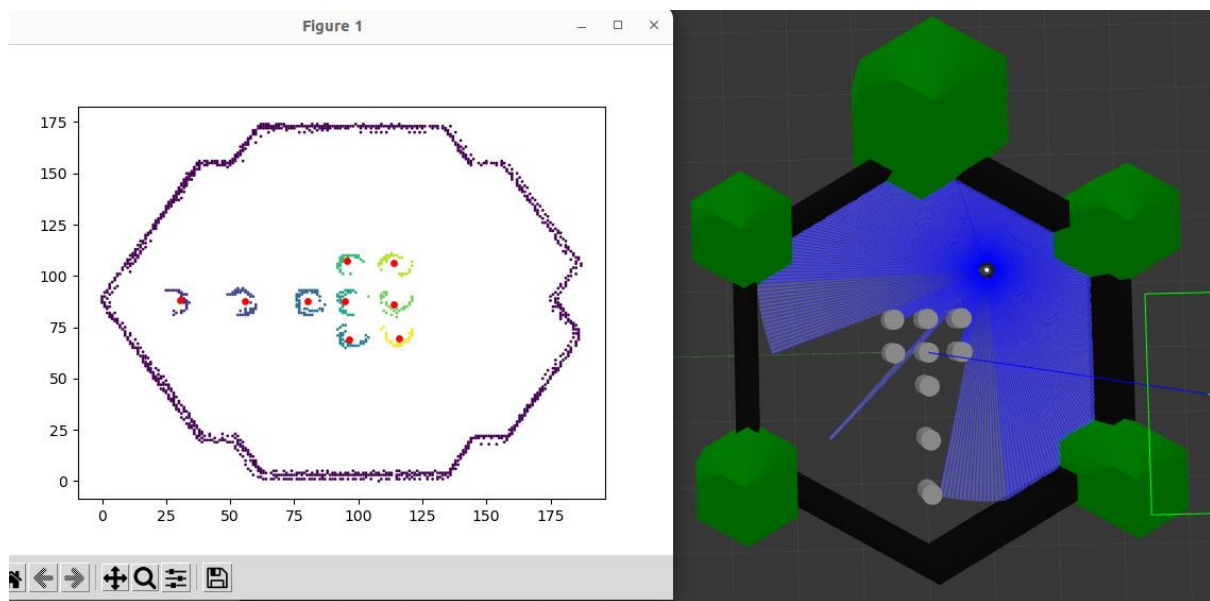


## Pacote de Movimentação e Contagem de Obstáculos em uma determinada área.

O objetivo desse pacote é resolver o desafio de contagem de objetos dentro de um determinado recinto, com um robô autônomo e zero interferência humana.

O Desafio consiste em utilizar um robô móvel com um conjunto de sensores em um ambiente parcialmente conhecido para contar a quantidade de obstáculos neste ambiente. O robô pode ser inicializado em qualquer lugar do ambiente de forma aleatória e pode mover-se livremente. O robô não deve ser teleoperado ou depender de comandos externos. O robô deve ser capaz de realizar a tarefa para uma quantidade de obstáculos variáveis posicionados de forma aleatória no ambiente.

Podemos ver o exemplo de funcionamento abaixo:



Para resolver este problema será utilizado o **ROS2 Humble** e o **Gazebo**.

### ROS2 Humble

Para a instalação do ROS2 Humble você deve fazer os seguintes passos:

- 1- `sudo apt install ros-humble-desktop`
- 2- `sudo apt install ros-humble-ros-base`
- 3- `sudo apt install ros-dev-tools`

Os três passos são responsáveis pela instalação do Humble, bibliotecas e pacotes de mensagens, além de um compilador para criar os pacotes utilizados no ROS.

### Gazebo

Para a instalação do Gazebo você deve fazer o seguinte passo:

1- `curl -sSL http://get.gazebo.org | sh`

Esse comando instala a última versão do Gazebo de qualquer distribuição do Linux.

## **Programa**

Para começar a executar os nossos códigos, primeiro é necessário colocar o turtlebot dentro do mundo `turtlebot3_world`.

*---> `ros2 launch turtlebot3_gazebo turtlebot3_world.launch.py`*

obs.: Para utilizar os mundos criados é substituído arquivo 'model.sdf' do mundo criado dentro da pasta 'models' do próprio turtlebot3.

Após isso, é necessário fazer o launch dos dois nós utilizados de movimentação e criação do mapa:

`turtlebot3_control.py` (Controle de movimentação)

`map_creator` (criação do mapa, detecção dos objetos e escolha de melhor local para ir)

*---> `ros2 launch tb3_ra_bringup tb3_ra_project.launch.py`*

De vez em quando é necessário utilizar o `teleop_keyboard` para fazer alguns ajustes:

*---> `ros2 run teleop_twist_keyboard teleop_twist_keyboard`*