Figure-8 Trajectory & Pen Control in ROS 2 with Turtlesim

Overview:

This project demonstrates ROS 2 concepts including custom nodes, publishers, subscribers, services, and launch files using the turtlesim simulator. The turtle moves in a continuous figure-8 pattern, and its pen can be toggled on or off through a service call.

Video Link:
1. https://drive.google.com/file/d/14SmwMv3whGuf6uPVDxCPPaETqx-yEdCB/view?usp=sharing

2. https://drive.google.com/file/d/161jji_Djhs2u1ddrNafGPwWulXYyCEZQ/view?usp=sharing

Key Components:

Node1: `figure8_driver`

Purpose: Makes the turtle move in a continuous **figure-8** pattern using linear and angular velocity commands.

Topics:

- Publishes to: `/turtle1/cmd_vel` (`geometry_msgs/msg/Twist`) to move the turtle.
- Subscribes to: `/turtle1/pose` (`turtlesim/msg/Pose`) to get live position and heading.

Key Features:

- **ROS 2 Parameters**: `pattern_speed` and `angular_speed_multiplier` control the figure-8 dynamics.
- **State Machine**: Alternates between `'turn_left'` and `'turn_right'` states using timers to approximate two loops of the 8.
- **Pose Logging**: Logs turtle's `(x, y, θ)` every second to show movement and orientation.

Node 2: trace_toggle

Purpose: Provides a **service interface** to turn the turtle's pen on or off, and moves the turtle slightly forward each time it's triggered.

Service:

- Implements: `/toggle_trace` (`std_srvs/srv/SetBool`)
- Internally calls: `/turtle1/set_pen` (`turtlesim/srv/SetPen`) to turn drawing ON or OFF.

Other Actions:

- Publishes to `/turtle1/cmd_vel` to nudge the turtle forward on toggle, making the effect immediately visible.
-

- **Launch File: bringup.launch.py**
  - Launches turtlesim_node, figure8_driver, and trace_toggle together

Usage Instructions

- Update System & Install Utilities

  sudo apt update && sudo apt upgrade -y

```
sudo apt install software-properties-common curl -y
```

- Add the ROS 2 GPG Key & Repository

```
sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg

echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null
```

- Install ROS 2 Humble Desktop

```
sudo apt update

sudo apt install ros-humble-desktop -y
```

- Source ROS in the Shell

```
echo "source /opt/ros/humble/setup.bash" >> ~/.bashrc

source ~/.bashrc
```

- Install colcon Build Tools

```
sudo apt install python3-colcon-common-extensions -y
```

- Build the Workspace

```
cd ~/ros2_ws

colcon build

source install/setup.bash
```

Running Figure8_driver Node:

Terminal 1:

```
ros2 run turtlesim turtlesim_node
```

Terminal 2:

source install/setup.bash

ros2 run turtle_control figure8_driver

Running TraceToggleService Node:

Terminal 2:

source install/setup.bash

ros2 run turtle_control trace_toggle

Terminal 3:

source install/setup.bash

ros2 service call /toggle_trace std_srvs/srv/SetBool "{data: false}" → pen off

ros2 service call /toggle_trace std_srvs/srv/SetBool "{data: true}" → pen on

Running Launch File:

Terminal 1:

source install/setup.bash

ros2 launch turtle_control bringup.launch.py

Terminal 2:

source install/setup.bash

ros2 service call /toggle_trace std_srvs/srv/SetBool "{data: false}" → pen off

ros2 service call /toggle_trace std_srvs/srv/SetBool "{data: true}" → pen on

Challenges Encountered

1. Service Not Responding or Not Found

- **Cause**: `trace_toggle_service` was not running or not included in the launch file initially.

2. Build Errors

- **Examples**:
    - `SyntaxError` in `setup.py` due to unmatched brackets.
    - `NameError: name 'os' is not defined` due to missing import.

3. Turtle Not Moving

- **Cause**: Missing `rclpy.spin()` or failure in publishing to `/cmd_vel`.

4. ROS Installation Issues in WSL

- **Cause**: Ubuntu 24.04 (“Noble”) does not have official ROS 2 packages yet.

5. Launching Issues

- **Problem**: Only one node (e.g., figure8) was running.
- **Fix**: Validated `bringup.launch.py` structure and ensured `entry_points` in `setup.py` were correctly configured.