MRT Assignment 1

Harshit Somani

October 16, 2024

1 Introduction

Completed the Episode 0 - A Study in ROS2 tutorial. All those work files are in the `src/week0_tutorials` directory.

This tutorial was used as an inspiration for solving the assignment.

2 Report

2.1 The Talker and Listener Scripts

The `talker` files the same as the code given in the tutorial with some minor modifications.

Here are the steps I took for the `talker` files:

1. Create 2 `talker` files: `talker1.py` and `talker2.py` and copy the default `talker` code from the tutorial.
2. Replace the lines

```
msg.data = 'Hello World'
node = rclpy.create_node('minimal_publisher')
publisher = node.create_publisher(String, 'topic', 10)
```

with

```
msg.data = 'You cannot hide. I see you.'
node = rclpy.create_node('publisher1')
publisher = node.create_publisher(String, 'listen_1', 10)
```

for `talker1.py` and

```
msg.data = 'Build me an army worthy of Mordor.'
node = rclpy.create_node('publisher2')
publisher = node.create_publisher(String, 'listen_2', 10)
```

for `talker2.py`.

The new topics are now `\listen1` and `\listen2` instead of `\topic`

For the `listener` file, copy the default `listener` code from the tutorial and make the following changes:

1. Replace the line

```
subscription = node.create_subscription(String, 'topic', listener_callback, 10)
```

with

```
subscription1 = node.create_subscription(String, 'listen_1', listener_callback1, 10)
subscription2 = node.create_subscription(String, 'listen_2', listener_callback2, 10)
```

This creates 2 subscriptions for the subscriber node. But instead of calling the same function `listener_callback`, they call two separate functions with different strings as parameters (since they are subscribed to different topics).

2. The two listener functions are:

```
def listener_callback1(msg):
print('%s ' % msg.data, end='')

def listener_callback2(msg):
print('%s' % msg.data)
```

`listener_callback1` prints the message from `listen_1` and doesn't add a newline after the print operation is finished (`end=''`).
`listener_callback2` prints the message from `listen_2` and adds a breaks the line after the print operation is finished (as usual).

3. Change `node = rclpy.create_node('minimal_subscriber')` to `node = rclpy.create_node('subscriber')` just for fun.

2.2 The setup.py File

The scripts of `listener.py, talker1.py, talker2.py` are given commands to run on console.

```
'console_scripts': [
    'publisher1 = sauron.talker1:main',
    'publisher2 = sauron.talker2:main',
    'subscriber = sauron.listener:main',
],
```

2.3 The Launcher File

The launcher file `pubsub_launch.py` in the `src/sauron/launch` directory is

```
from launch import LaunchDescription
from launch_ros.actions import Node

def generate_launch_description():
    return LaunchDescription([
        Node(
            package='sauron',
            executable='publisher1',
        ),
        Node(
            package='sauron',
            executable='publisher2',
        ),
        Node(
            package='sauron',
            executable='subscriber',
        ),
    ])
```

Adding the line

```
(os.path.join('share', package_name, 'launch'), glob(os.path.join('launch', 'pubsub_launch.py'))),
```

to `setup.py` in the `data_files` allows us to launch all three nodes using the `ros2 launch sauron pubsub_launch.py` command in console.