IIROS 2 ■ Contract IIROS 2

[08 – ROS2 Parameter & Logs]

한국폴리텍대학교 성남캠퍼스

1 Homework Review



Homework Review

Mission

- turtlesim_node를 실행시킨 후 거북이는 (x: 1, y: 1, theta:0)으로 위치
- 사용자가 지정한 영역 (x, y, x_l, y_l) → (x, y) (x, y+y_l), (x+x_l, y) (x+x_l, y+y_l) 으로 이동 하도록 동작
 - 조건1) 거북이는 랜덤주행
 - 조건2) action 방식을 이용하여 사용자가 도착영역을 goal로 지정하면 주행시작
 - 조건3) 도착영역에 도달할 때 까지 현재 좌표를 피드백으로 전송
 - 조건4) 도착영역에 도달하면 거북이 주행 종료



0

Homework Review

- ❖ Mission 다음 조건을 Multi-Thread로 작성 하시오.
 - 1. Multi-thread 사용
 - 2. action을 이용하여 설계
 - 3. turtlesim_node 내에 거북이가 지나갈 수 없는 장애물 영역을 3개 랜덤하게 설정 (직사각형으로 제한, 거북이가 지나갈 수 있는 경로가 있으면 크기 무관)
 - 4. 좌측 하단 영역(x: 1, y: 1) 에서 출발하여 우측상단 영역에 도착할 수 있도록 자율 주행 프로그램 작성

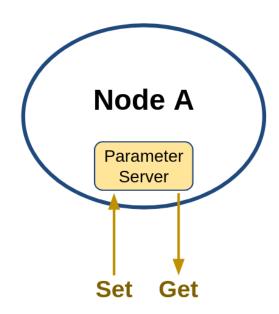


5. 가능한 최단 시간에 도달할 수 있도록 소프트웨어 설계





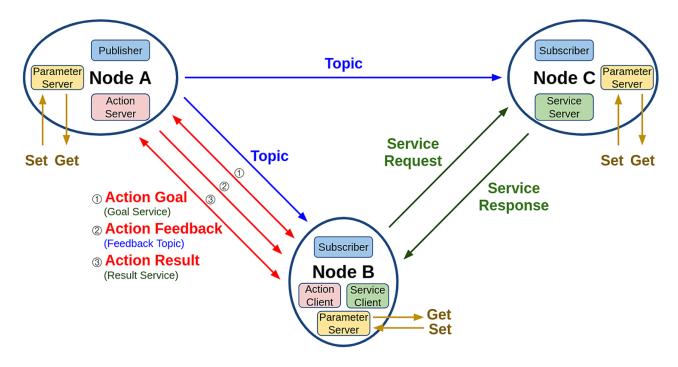
- C#의 Property와 유사한 개념
- C#에서 내부 데이터 필드의 접근을 get/set 접근자를 이용하여 읽고/쓰기 실행
- 이와 유사하게 ROS2의 노드 내부의 데이터 필드도 Parameter라는 개념으로 get/set을 통해 읽고 쓸 수 있음







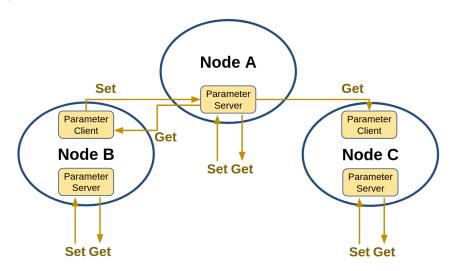
- 각 노드의 Parameter server를 실행시켜 외부의 Parameter client 간의 통신으로 파라미터를 변경하는 것으로 서비스와 동일
- 단 노드 내 매개변수 또는 글로벌 매개변수를 서비스 통신 방법을 사용하여 외부에서 Set/Get 할 수 있게 함







- 파라미터 관련 기능은 RCL(ROS Client Libraries)의 기본 기능으로 모든 노드가 자신만의 Parameter server를 가지고 있음
- 각 노드는 Parameter client도 포함시킬 수 있어서 자기 자신의 파라미터 및 다른 노드의 파라미터를 읽고 쓸 수 있음
- 추가 프로그래밍이나 컴파일 없이 능동적으로 변화 가능한 프로세스 생성 가능
- 각 파라미터는 yaml 파일 형태의 파라미터 설정 파일로 되어 있음







Parameter Example

• turtlesim_node 및 turtle_teleop_key 노드를 각각 실행 후 parameter list 조회

```
daesung@DSThinkPad:~$ ros2 param list
/teleop_turtle:
  qos_overrides./parameter_events.publisher.depth
  gos_overrides./parameter_events.publisher.durability
  qos_overrides./parameter_events.publisher.history
  gos_overrides./parameter_events.publisher.reliability
  scale_angular
  scale linear
  use_sim_time
/turtlesim:
  background_b
  background_g
  background_r
  qos_overrides./parameter_events.publisher.depth
  qos_overrides./parameter_events.publisher.durability
  gos_overrides./parameter_events.publisher.history
  gos_overrides./parameter_events.publisher.reliability
  use_sim_time
daesung@DSThinkPad:~$
```

```
daesung@DSThinkPad:~$ ros2 run turtlesim turtle_teleop_key
Reading from keyboard
-------
Use arrow keys to move the turtle.
Use G|B|V|C|D|E|R|T keys to rotate to absolute orientations. 'F' to cancel a rotation.
'Q' to quit.
```



1

- Parameter Example
 - 현재 실행 중인 노드 2개의 정보가 나타남을 확인할 수 있음

```
daesung@DSThinkPad:~$ ros2 param list
/teleop_turtle:
 qos_overrides./parameter_events.publisher.depth
 qos_overrides./parameter_events.publisher.durability
 qos_overrides./parameter_events.publisher.history
 qos_overrides./parameter_events.publisher.reliability
 scale_angular
 scale_linear
 use_sim_time
/turtlesim:
 background_b
 background_g
 background_r
 qos_overrides./parameter_events.publisher.depth
 qos_overrides./parameter_events.publisher.durability
 qos_overrides./parameter_events.publisher.history
 qos_overrides./parameter_events.publisher.reliability
 use_sim_time
daesung@DSThinkPad:~$
```



1

- ❖ Parameter Example 배경화면 색 : 정보 조회
 - turtlesim_node의 배경색 parameter 조회

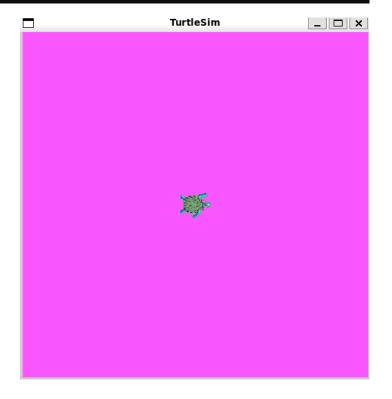
```
daesung@DSThinkPad:~$ ros2 param list
/teleop_turtle:
 gos_overrides./parameter_events.publisher.depth
 gos_overrides./parameter_events.publisher.durability
 gos_overrides./parameter_events.publisher.history
 gos_overrides./parameter_events.publisher.reliability
 scale_angular
 scale_linear
turtlesim:
 background_b
 background_g
  background_r
 gos_overrides./parameter_events.publisher.depth
 gos_overrides./parameter_events.publisher.durability
 qos_overrides./parameter_events
                            daesung@DSThinkPad:~$ ros2 param get /turtlesim background_r
 gos_overrides./parameter_
                           Integer value is: 69
 use_sim_time
                            daesung@DSThinkPad:~$ ros2 param get /turtlesim background_g
daesung@DSThinkPad:~$
                           Integer value is: 86
                            daesung@DSThinkPad:~$ ros2 param get /turtlesim background_b
                           Integer value is: 255
                            daesung@DSThinkPad:~$
```



❖ Parameter Example – 배경화면 색 : 변경

• turtlesim_node의 배경색 parameter 설정

```
daesung@DSThinkPad:~$ ros2 param set /turtlesim background_r 250
Set parameter successful
daesung@DSThinkPad:~$
```







- ❖ Parameter Example 배경화면 : 변경정보 저장
 - 노드 실행 시 기본 설정 parameter 값과 다르게 사용자 설정 값을 저장하고 싶은 경우가 있음
 - 이러한 경우에 parameter 전체를 저장할 수 있음
 - 이를 위해서 사용하고자 하는 Package 폴더에 params 폴더 생성 후 ~~~.yaml

파일을 생성

```
daesung@DSThinkPad:~/ros2_work/src/first_pkg$ tree
    first_pkg
       action_server.py
        dist_action_server.py
        multi_thread_test.py
        srv_server.py
        t_cmd_client.py
        topic_publisher.py
        topic_subscriber.py
        t_rndmv_mnq.py
        turtle_cmd_pose.py
      - turtlesim.yaml
   setup.cfq
    setup.py
      - test_copyright.py

    test_flake8.py

      test_pep257.py
4 directories, 18 files
daesung@DSThinkPad:~/ros2_work/src/first_pkg$
```





- ❖ Parameter Example 배경화면 : 변경정보 저장
 - turtlesim.yaml 파일 정보 확인





- ❖ Parameter Example 배경화면 : 변경정보 수정
 - VS code를 실행하여 turtlesim.yaml 파일 정보 수정

background_b: 255

background_g: 86

background_r: 250



background b: 203

background_g: 192

background_r: 255



- ❖ Parameter Example 배경화면 : 변경정보 수정
 - 저장된 turtlesim.yaml 파일 정보를 turtlesim으로 불러오기

ros2 param load /turtlesim turtlesim.yaml

```
daesung@DSThinkPad:~/ros2_work/src/first_pkg/params$ ros2 param load /turtlesim turtlesim.yaml
Set parameter background_b successful
Set parameter background_g successful
Set parameter background_r successful
                                                                           TurtleSim
                                                                                         Set parameter gos_overrides./parameter_events.publisher.de
                                                                                                /parame
ter_events.publisher.depth' cannot be set because it is rea
Set parameter gos_overrides./parameter_events.publisher.du
                                                                                                ides./p
arameter_events.publisher.durability' cannot be set because
Set parameter qos_overrides./parameter_events.publisher.his
                                                                                                s./para
meter_events.publisher.history' cannot be set because it is
Set parameter qos_overrides./parameter_events.publisher.re
                                                                                                rides./
parameter_events.publisher.reliability' cannot be set becau
Set parameter use_sim_time successful
daesung@DSThinkPad:~/ros2_work/src/first_pkg/params$
```

K

- ❖ Python 코드에서 Parameter 접근하기
 - Action Server에서 사용한 코드(dist_action_server.py)에 파라미터 관련 코드 추가

```
import rclpy as rp
from rclpy.action import ActionServer
from rclpy.executors import MultiThreadedExecutor
from rclpy.node import Node
from turtlesim.msg import Pose
from geometry msgs.msg import Twist
from first msg.action import ActionData
from first pkg.topic subscriber import TurtlesimSubscriber
from rcl interfaces.msg import SetParametersResult
import math
import time
class Sub Action(TurtlesimSubscriber):
    def __init__(self, ac_server):
        super(). init ()
        self.ac server = ac server
    def callback(self, msg):
        self.ac_server.cur_pose = msg
```

1

- ❖ Python 코드에서 Parameter 접근하기
 - Action Server에서 사용한 코드(dist_action_server.py)에 파라미터 관련 코드 추가

```
class DistServer(Node):
   def init (self):
       super().__init__('dist_action server')
       self.t dist = 0
       self.first time = True
       self.cur_pose = Pose()
       self.pre pose = Pose()
       self.pub = self.create publisher(Twist, '/turtle1/cmd vel', 10)
       self.act srv = ActionServer(self, ActionData, 'dist act srv', self.exe callback)
        self.declare parameter('quatile time', 0.75)
        self.declare parameter('almost goal time', 0.95)
        (g quatile time, g almosts time) = self.get parameters(['quatile time',
'almost goal time'])
        self.quatile time = g quatile time.value
        self.almosts time = g almosts time.value
        self.add on set parameters callback(self.parameter callback)
```

❖ Python 코드에서 Parameter 접근하기

----- 이후 추가사항 없음 ------

• Action Server에서 사용한 코드(dist_action_server.py)에 파라미터 관련 코드 추가

```
def parameter_callback(self, params):
    for param in params:
        print(param.name, ' is changed to ', param.value)

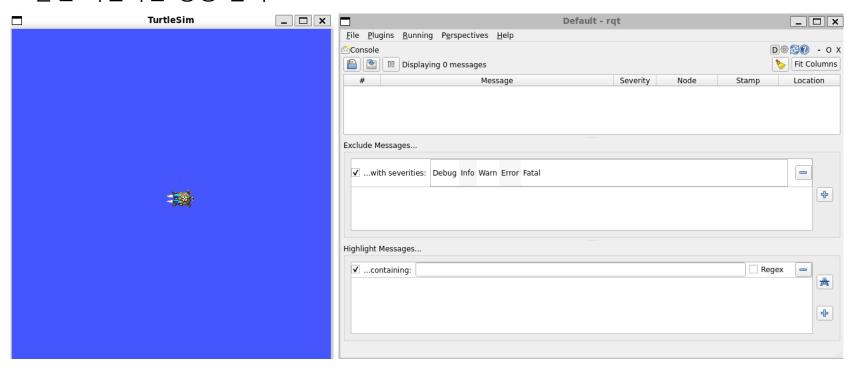
    if param.name == 'quatile_time':
        self.quatile_time = param.value
    if param.name == 'almost_goal_time':
        self.almosts_time = param.value
    print('quatile_time and almost_goal_time is ', self.quatile_time, self.almosts_time)
    return SetParametersResult(successful=True)
```

:::ROS

- ❖ print() → log
 - 현재까지는 중간의 데이터 처리과정 확인을 위해서 print 문을 이용해서 데이터 확인
 - 하지만 대부분의 ROS 유저들은 log에 기록을 남기면서 정상인 정보, 경고, 에러 등등을 수준별로 관리
 - 의도한 동작이 되지 않는 상황을 점검할 때 유용한 도구가 로그임
- 시간정보와 기록된 로그 정보를 이용해 보다 효율적으로 로봇을 점검할 수 있음
- 로그에 대해 살펴볼 예정



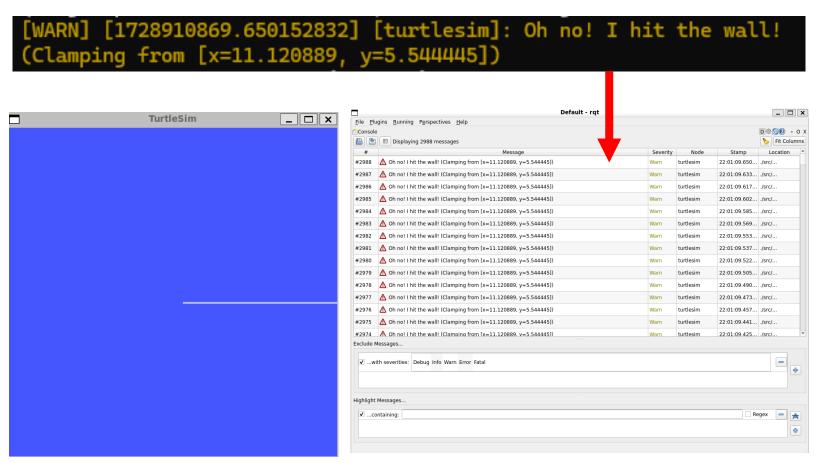
- ❖ rqt_console을 이용해 로그 확인
 - turtlesim_node 실행 후 rqt 실행 → Plugins에서 Logging의 Console 선택 후 1초에 한번 직진하는 명령 입력



ros2 topic pub -r 1 /turtle1/cmd_vel geometry_msgs/msg/Twist "{linear: {x:
2, y:0, z: 0}, angular:{x: 0, y: 0, z: 0}}"



- ❖ rqt_console을 이용해 로그 확인
 - 1초 마다 직진하는 명령 입력 후 거북이가 화면을 벗어나면 다음과 같은 메시지가 출력됨. Logging console은 조건이나 상황을 설정해 원하는 메시지를 관찰할 수 있음





- ❖ 로그 메시지 직접 만들기
 - 마지막 파라미터와 관련 내용을 추가한 소스코드를 대상으로 로그 레벨 info를 기록하는 코드를 추가
 - logger 함수를 이용해 info 레벨로 문자열을 로그로 기록. 로그에 기록되고, 터미널에도 출력 됨

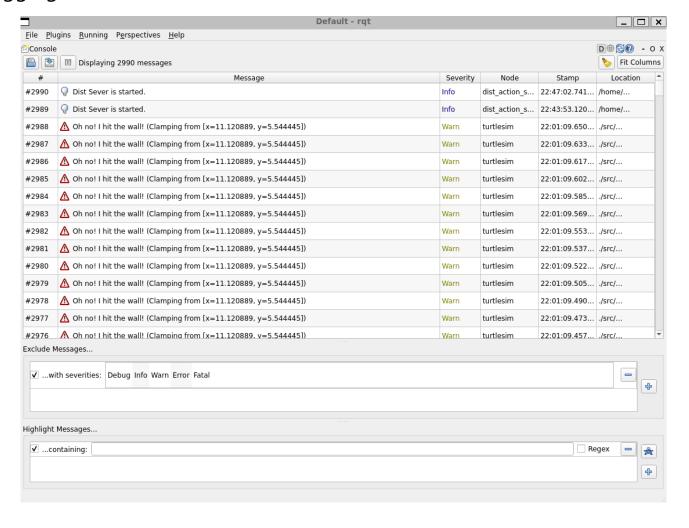
```
self.get_logger().info('Dist Sever is started.')
```

```
class DistServer(Node):
    def __init__(self):
        super().__init__('dist_action_server')
        self.t_dist = 0
        self.first_time = True
        self.cur_pose = Pose()
        self.pre_pose = Pose()
        self.pub = self.create_publisher(Twist, '/turtle1/cmd_vel', 10)
        self.act_srv = ActionServer(self, ActionData, 'dist_act_srv', self.exe_callback)
        self.get_logger().info('Dist Sever is started.')

        self.declare_parameter('quatile_time', 0.75)
        self.declare_parameter('almost_goal_time', 0.95)
```

daesung@DSThinkPad:~/ros2_work\$ ros2 run first_pkg dist_action_server_parms [INFO] [1728913433.120256275] [dist_action_server]: Dist Sever is started.

- ❖ 로그 메시지 직접 만들기
 - Logging console에도 해당 내용 표시



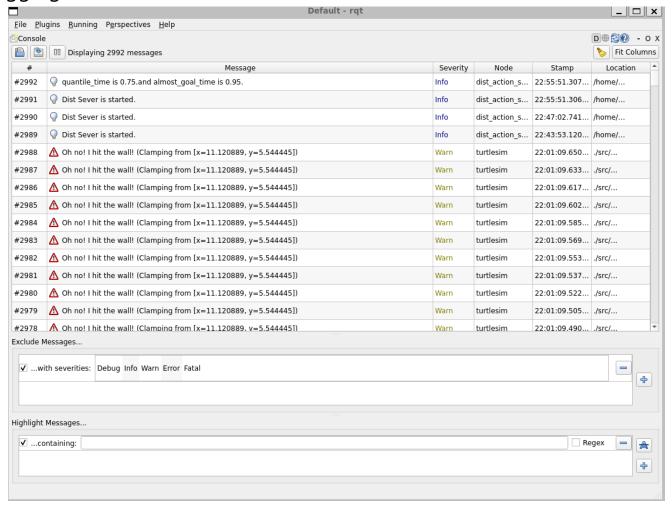


- ❖ 로그 메시지 직접 만들기
 - 다음과 같은 위치에 로그 정보를 출력하는 코드를 추가

```
output msg = 'quantile time is ' + str(self.quatile time) + '.'
     output_msg = output_msg + 'and almost_goal_time is ' + \
                    str(self.almosts_time) + '.'
     self.get logger().info(output_msg)
self.declare parameter('quatile time', 0.75)
self.declare parameter('almost goal time', 0.95)
(g quatile time, g almosts time) = self.get parameters(['quatile time', 'almost goa
self.quatile time = g quatile time.value
self.almosts time = g almosts time.value
output msg = 'quantile time is ' + str(self.quatile time) + '.'
output_msg = output_msg + 'and almost_goal_time is ' + str(self.almosts_time) +
self.get_logger().info(output_msg)
self.add_on_set_parameters_callback(self.parameter_callback)
daesung@DSThinkPad:~/ros2_work$ ros2 run first_pkg dist_action_server_parms_log
[INFO] [1728914151.306761427] [dist_action_server]: Dist Sever is started.
[INFO] [1728914151.307342073] [dist_action_server]: quantile_time is 0.75.and almost_goa
l_time is 0.95.
```



- ❖ 로그 메시지 직접 만들기
 - Logging console에도 해당 내용 표시





- ❖ 로그 메시지 직접 만들기
 - 다음과 같은 위치에 로그 정보를 출력하는 코드를 추가

```
output msg = 'quantile time is ' + str(self.quatile time) + '.'
     output_msg = output_msg + 'and almost_goal_time is ' + \
                    str(self.almosts_time) + '.'
     self.get logger().info(output_msg)
self.declare parameter('quatile time', 0.75)
self.declare parameter('almost goal time', 0.95)
(g quatile time, g almosts time) = self.get parameters(['quatile time', 'almost goa
self.quatile time = g quatile time.value
self.almosts time = g almosts time.value
output msg = 'quantile time is ' + str(self.quatile time) + '.'
output_msg = output_msg + 'and almost_goal_time is ' + str(self.almosts_time) +
self.get logger().info(output msg)
self.add_on_set_parameters_callback(self.parameter_callback)
daesung@DSThinkPad:~/ros2_work$ ros2 run first_pkg dist_action_server_parms_log
[INFO] [1728914151.306761427] [dist_action_server]: Dist Sever is started.
[INFO] [1728914151.307342073] [dist_action_server]: quantile_time is 0.75.and almost_goa
l_time is 0.95.
```

- ❖ 로그 메시지 직접 만들기
 - 다음과 같은 print 코드를 앞에서 작성했던 로그 정보로 저장 및 출력하는 코드로 수정

```
def parameter_callback(self, params):
    for param in params:
        print(param.name, ' is changed to ', param.value)

    if param.name == 'quatile_time':
        self.quatile_time = param.value
    if param.name == 'almost_goal_time':
        self.almosts_time = param.value

print('quatile_time and almost_goal_time is ', self.quatile_time, self.almosts_time)

return SetParametersResult(successful=True)
```

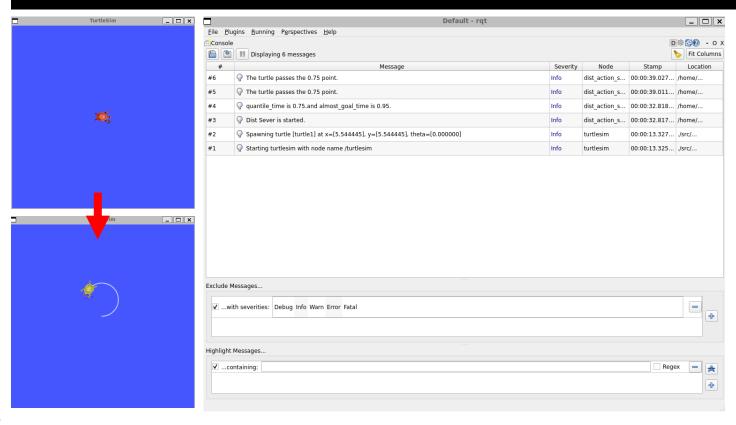


- ❖ 로그 메시지 직접 만들기
- 다음과 같이 exe_callback() 함수 내부 수정
- 목표지점까지 남은 거리가 quatile time 비율에 근접하면 로그를 남기도록 프로그래밍

```
while True:
    self.t dist += self.cal diff pose()
    feedback msg.remained dist = goal handle.request.dist - self.t dist
    goal handle.publish feedback(feedback msg)
    self.pub.publish(msg)
    tmp = abs(feedback msg.remained dist - goal handle.request.dist * self.quatile time)
    if tmp <0.02:
        output msg = 'The turtle passes the ' + str(self.quatile time) + ' point. '
        self.get logger().info(output msg)
    time.sleep(0.01)
    if feedback msg.remained dist < 0.01:
        break
    tmp = abs(feedback_msg.remained_dist - goal_handle.request.dist * self.quatile_time)
    if tmp <0.02:
         output_msg = 'The turtle passes the ' + str(self.quatile_time) + ' point. '
         self.get logger().info(output msg)
```

- ❖ 로그 메시지 직접 만들기
- turtlesim_node 실행. rqt log console과 로그 기능을 추가한 코드 실행 후 다음 명령어를 실행

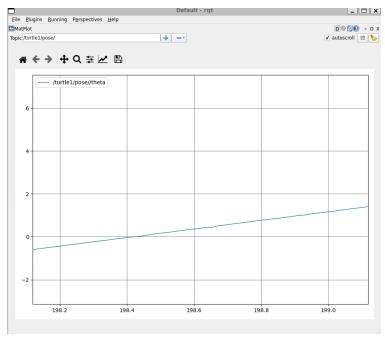
ros2 action send_goal /dist_act_srv first_msg/action/ActionData '{linear_x:
2., angular_z: 2., dist: 2.}'





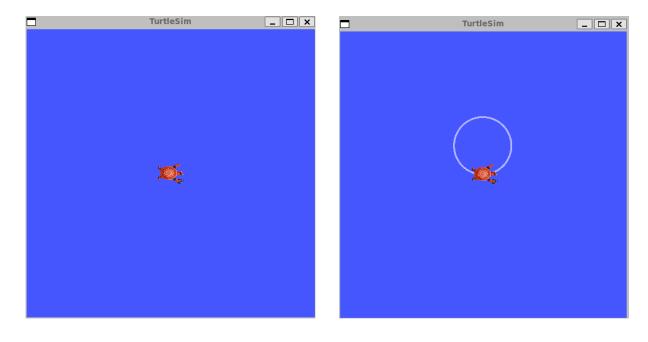
- rqt_plot
- rqt plugins 중 visualization의 plot 선택, topic은 turtle1/pose/ 로 선택
- 선택 후 필요 없는 값은 제거(theta 값만 남기도록 함)
- 다음 명령 실행 후 변화되는 theta 값 확인

ros2 action send_goal /dist_act_srv first_msg/action/ActionData '{linear_x:
2., angular_z: 2., dist: 2.}'



- rosbag
- topic을 기록하는 용도로 사용
- 데이터의 변화 관찰 및 디버그로 활용할 수 있음
- rosbag 확인을 위해 다음과 같은 준비

```
ros2 topic pub -r 1 /turtle1/cmd_vel geometry_msgs/msg/Twist "{linear: {x:
2., y: 0., z: 0.}, angular: {x: 0., y: 0., z: 1.8}}"
```





- rosbag
- 새로운 터미널에 다음과 같은 명령 입력
 - ✓ -o 옵션은 이름 지정
 - ✓ -a 옵션은 모든 토픽을 기록, 원하는 토픽 이름만 지정하여 기록할 수 있음

ros2 bag record -o turtle_test -a

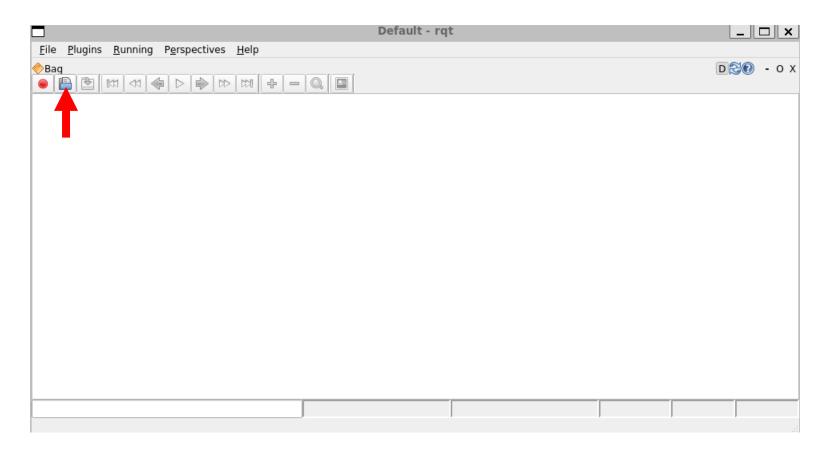
```
daesung@DSThinkPad:~$ ros2 bag record -o turtle_test -a
[INFO] [1728919710.576842755] [rosbaq2_recorder]: Press SPACE for pausing/resuming
[INFO] [1728919710.580227423] [rosbag2_storage]: Opened database 'turtle_test/turtle_tes
t_0.db3' for READ_WRITE.
[INFO] [1728919710.582373087] [rosbag2_recorder]: Listening for topics...
[INFO] [1728919710.582388114] [rosbag2_recorder]: Event publisher thread: Starting
[WARN] [1728919710.583608048] [ROSBAG2_TRANSPORT]: Hidden topics are not recorded. Enabl
e them with --include-hidden-topics
[INFO] [1728919710.584560322] [rosbag2_recorder]: Subscribed to topic '/turtle1/pose'
[INFO] [1728919710.585293623] [rosbag2_recorder]: Subscribed to topic '/turtle1/color_se
nsor'
[INFO] [1728919710.586083195] [rosbaq2_recorder]: Subscribed to topic '/turtle1/cmd_vel'
[INFO] [1728919710.586674965] [rosbag2_recorder]: Subscribed to topic '/rosout'
[INFO] [1728919710.587130370] [rosbag2_recorder]: Subscribed to topic '/parameter_events
[INFO] [1728919710.587653263] [rosbag2_recorder]: Subscribed to topic '/events/write_spl
[INFO] [1728919710.587732422] [rosbag2_recorder]: Recording...
[INFO] [1728919739.108278167] [rosbag2_cpp]: Writing remaining messages from cache to th
e bag. It may take a while
[INFO] [1728919739.108996358] [rosbag2_recorder]: Event publisher thread: Exiting
[INFO] [1728919739.109368104] [rosbag2_recorder]: Recording stopped
daesung@DSThinkPad:~$ ls
ROS2 ros2_work tmp turtle_test
```

- rosbag
- turtlesim_node를 다시 실행하고 ros2 bag play 실행

ros2 bag play turtle_test/

```
daesung@DSThinkPad:~$ ros2 bag play turtle_test/
[INFO] [1728919965.384936651] [rosbag2_storage]: Opened database 'turtle_test/turtle_tes
t_0.db3' for READ_ONLY.
[INFO] [1728919965.385007828] [rosbag2_player]: Set rate to 1
[INFO] [1728919965.393977596] [rosbag2_player]: Adding keyboard callbacks.
[INFO] [1728919965.394026285] [rosbag2_player]: Press SPACE for Pause/Resume
[INFO] [1728919965.394042690] [rosbag2_player]: Press CURSOR_RIGHT for Play Next Message
[INFO] [1728919965.394058636] [rosbag2_player]: Press CURSOR_UP for Increase Rate 10%
[INFO] [1728919965.394063197] [rosbag2_player]: Press CURSOR_DOWN for Decrease Rate 10%
[INFO] [1728919965.394391803] [rosbag2_storage]: Opened database 'turtle_test/turtle_test_0.db3' for READ_ONLY.
daesung@DSThinkPad:~$
```

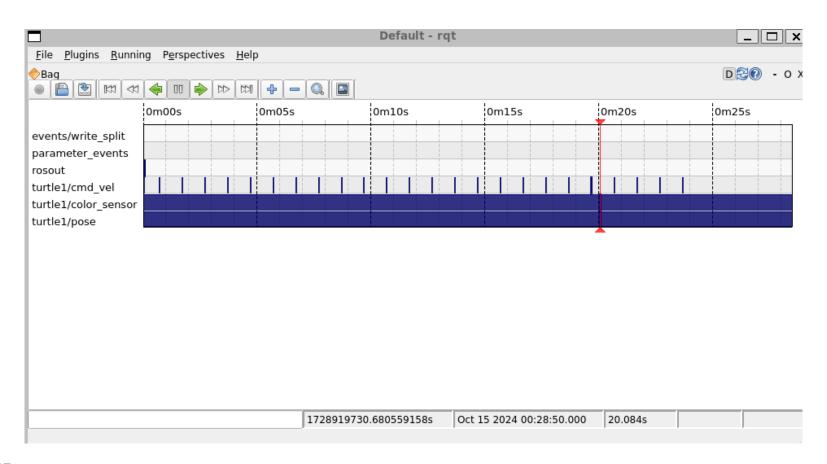
- rosbag
- rqt 실행 후 plugins에서 Logging의 Bag 선택
- 아래와 같이 logbag 선택 후 turtle_test 선택





rosbag

• 다음과 같이 나타난 logbag 차트에 오른쪽 마우스를 클릭하고 publish에 대한 옵션을 선택하면 turtlesim_node가 기록된 topic에 의해 동작





roslaunch

• 지금까지 하나의 터미널에 하나의 노드만 실행

• launch 명령어는 여러 노드를 실행하거나 손쉽게 노드와 파라미터의 설정 등을 지정할 수 있음

• package 폴더에 launch 폴더 생성 후 ~~~.launch.py 파일 생성

roslaunch

- ~~~~.launch.py 파일에 다음과 같이 입력
- turtlesim 패키지의 turtlesim_node 와
 first_pkg의 topic_publisher 실행 지정

```
> resource
from launch import LaunchDescription
                                                        > test
from launch ros.actions import Node
                                                        package.xml
                                                        setup.cfg
def generate launch description():
                                                        setup.py
    return LaunchDescription(
            Node(
                namespace='', package='turtlesim',
                executable='turtlesim_node', output='screen'
            ),
            Node(
                namespace='', package='first_pkg',
                executable='topic_publisher', output='screen'
            ),
```

✓ FIRST_PKG [WSL: UBUNTU-22.04]

turtlesim_teleop.launch.py

> first_pkg

√ launch

> params

- roslaunch
- setup.py에 다음과 같이 추가 후 빌드

```
import os
import glob
setup(
    name=package_name,
    version='0.0.0',
    packages=find_packages(exclude=['test']),
    data_files=[
        ('share/ament_index/resource_index/packages',
            ['resource/' + package_name]),
        ('share/' + package_name, ['package.xml']),
        ('share/' + package_name + '/launch',glob.glob(os.path.join('launch',
'*.launch.py'))),
```



roslaunch

• launch 실행 결과 turtlesim_node가 실행되고 거북이가 회전

```
ros2 launch first_pkg turtlesim_teleop.launch.py
```

```
daesung@DSThinkPad:~/ros2_work$ ros2 launch first_pkg turtlesim_teleop.launch.py
[INFO] [launch]: All log files can be found below /home/daesung/.ros/log/2024-10-15-10-23-59-
501032-DSThinkPad-16745
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [turtlesim_node-1]: process started with pid [16752]
[INFO] [topic_publisher-2]: process started with pid [16754]
[turtlesim_node-1] QStandardPaths: wrong permissions on runtime directory /run/user/1000/, 07
55 instead of 0700
[turtlesim_node-1] [INFO] [1728955439.571383123] [turtlesim]: Starting turtlesim with node na
me /turtlesim
[turtlesim_node-1] [INFO] [1728955439.575250435] [turtlesim]: Spawning turtle [turtle1] at x=
[5.5444445], y=[5.5444445], theta=[0.000000]
```



roslaunch

parameter 지정을 위한 파일 추가 – dist_action.launch.py 후 빌드

```
from launch import LaunchDescription
from launch ros.actions import Node
def generate launch description():
    my launch = LaunchDescription()
    turtlesim node = Node(
        package='turtlesim', executable='turtlesim_node', output='screen',
        parameters=[
            {'background_r': 255},
            {'background_g': 192},
            {'background_b': 203},
    dist action node = Node(
        package='first_pkg', executable='dist_action_server', output='screen'
    my launch.add action(turtlesim node)
    my_launch.add_action(dist_action node)
    return my_launch
```

- roslaunch
- 빌드 후 launch를 실행하고 server에 명령어 전송

```
ros2 launch first_pkg turtlesim_teleop.launch.py
```

```
daesung@DSThinkPad:~/ros2_work$ ros2 launch first_pkg dist_action.launch.py
[INFO] [launch]: All log files can be found below /home/daesung/.ros/log/2024-10-15-10-27-26-
223132-DSThinkPad-17389
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [turtlesim_node-1]: process started with pid [17390]
[INFO] [dist_action_server-2]: process started with pid [17392]
[turtlesim_node-1] QStandardPaths: wrong permissions on runtime directory /run/user/1000/, 07
55_instead_of_0700
[turtlesim_node-1] QStandardPaths: wrong permissions on runtime directory /run/user/1000/, 07
55_instead_of_0700
[turtlesim_node-1] QStandardPaths: wrong permissions on runtime directory /run/user/1000/, 07
55_instead_of_0700
[turtlesim_node-1] QStandardPaths: wrong permissions on runtime directory /run/user/1000/, 07
55_instead_of_0700
[turtlesim_node-1] QStandardPaths: wrong permissions on runtime directory /run/user/1000/, 07
55_instead_of_0700
[turtlesim_node-1] QStandardPaths: wrong permissions on runtime directory /run/user/1000/, 07
55_instead_of_0700
[turtlesim_node-1] QStandardPaths: wrong permissions on runtime directory /run/user/1000/, 07
55_instead_of_0700
[turtlesim_node-1] QStandardPaths: wrong permissions on runtime directory /run/user/1000/, 07
55_instead_of_0700
[turtlesim_node-1] QStandardPaths: wrong permissions on runtime directory /run/user/1000/, 07
55_instead_of_0700
[turtlesim_node-1] QStandardPaths: wrong permissions on runtime directory /run/user/1000/, 07
55_instead_of_0700
[turtlesim_node-1] QStandardPaths: wrong permissions on runtime directory /run/user/1000/, 07
55_instead_of_0700
[turtlesim_node-1] QStandardPaths: wrong permissions on runtime directory /run/user/1000/, 07
55_instead_of_0700
[turtlesim_node-1] QStandardPaths: wrong permissions on runtime directory /run/user/1000/, 07
55_instead_of_0700
[turtlesim_node-1] QStandardPaths: wrong permissions on runtime directory /run/user/1000/, 07
55_instead_of_0700
[turtlesim_node-1] QStandardPaths: wrong permissions on runtime directory /run/user
```



A