ROS2 완독챌린지

9. 패키지 설치와 노드 실행

노드 (node): 하나의 실행 가능한 프로그램

패키지 : 하나 이상의 노드 또는 노드 실행을 위한 정보 등의 집합

메타패키지 : 패키지의 모음

ros2 패키지 조회

```
$ ros2 pkg list
```

ros2 패키지 중 turtlesim 패키지의 노드 조회

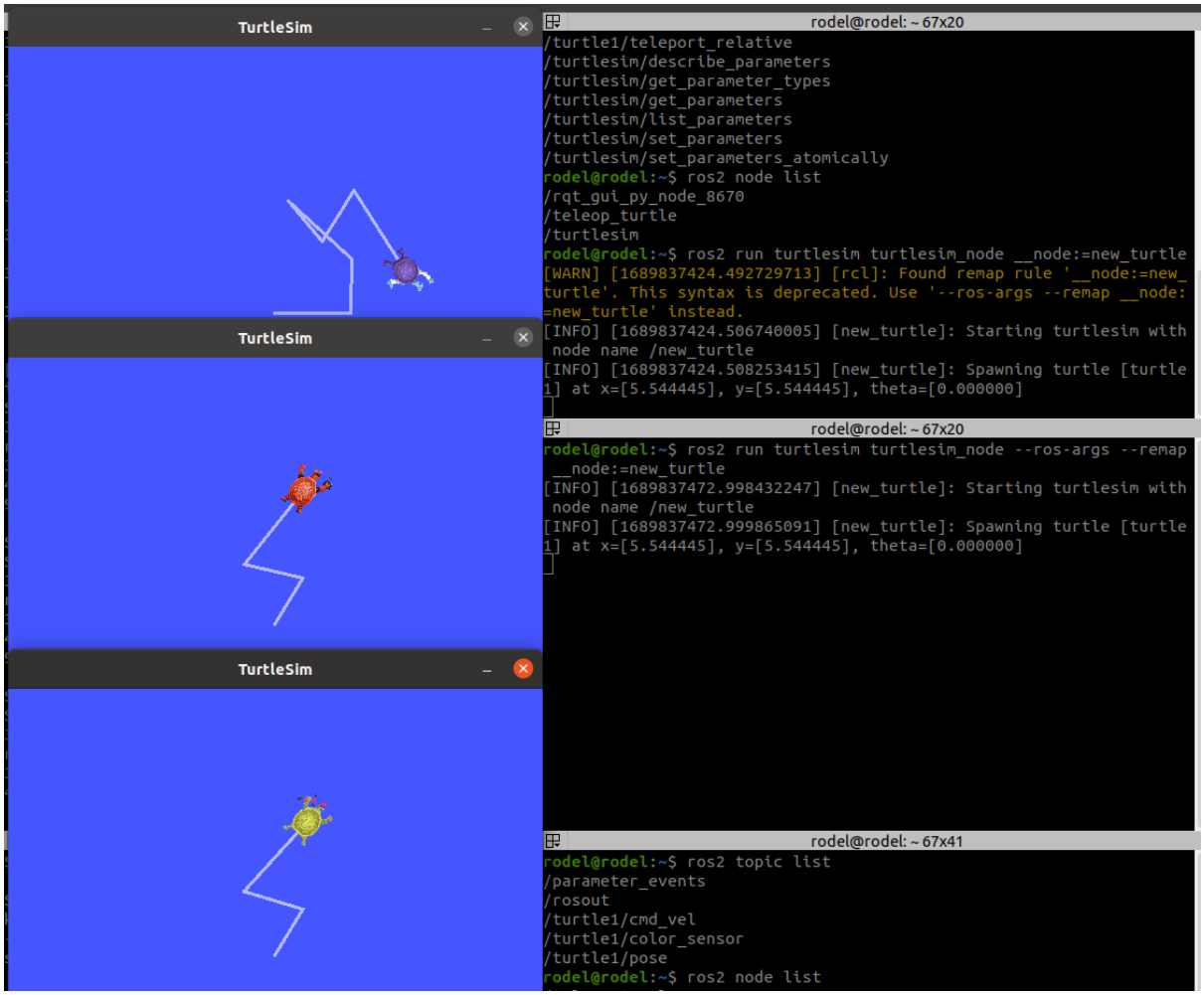
```
~$ ros2 pkg executables turtlesim
turtlesim draw_square
turtlesim mimic
turtlesim turtle_teleop_key
turtlesim turtlesim_node
```

10. ROS2 노드와 데이터 통신

ros2 run (패키지) (노드이름) : 노드 실행

```
$ ros2 run turtlesim turtlesim_node
$ ros2 run turtlesim turtle_teleop_key
```

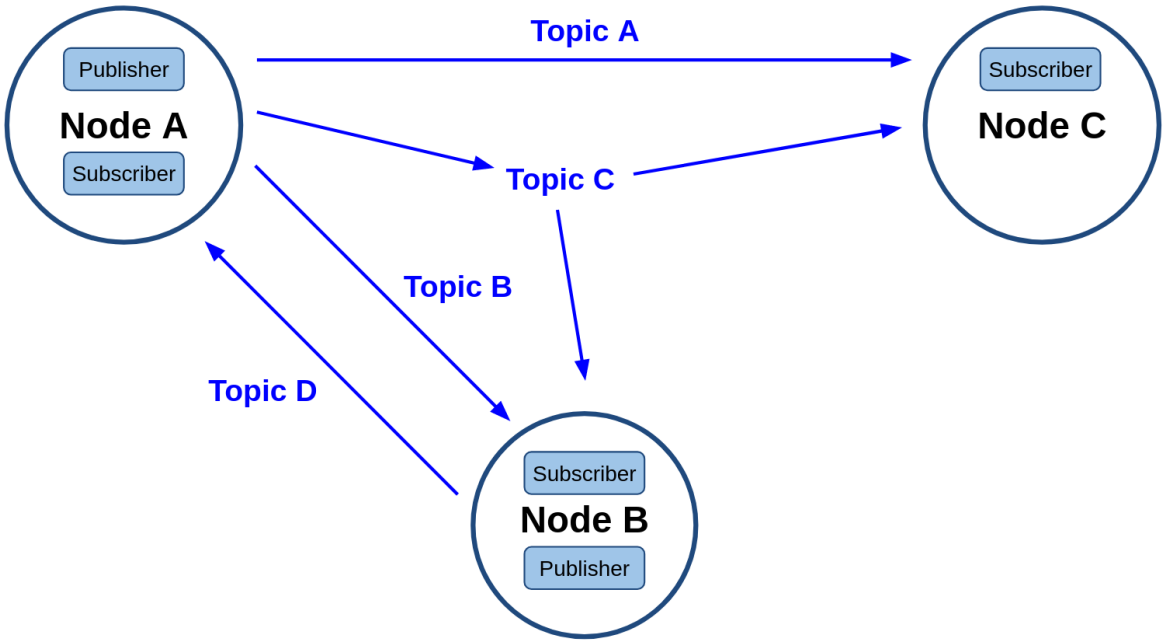
ros2 run turtlesim turtlesim_node



- turtlesim 패키지 속 turtlesim_node라는 노드 실행
- 동일 노드 복수 개 실행시키려면 같은 실행 코드 입력은 별로안좋음

→ foxy 부터는 —ros args —remap __node:=new turtle 의 형식으로.

11. 토픽



정의 : 비동기식 단방향 메시지 송수신 방식

Publisher : msg 인터페이스 형태의 메시지 발행 (Like 잡지 발행)

Subscriber : 메시지를 구독 (발행한 잡지를 선택해 받겠다고 구독)

토픽 A - 1:1 통신

토픽 C - 1:N 통신

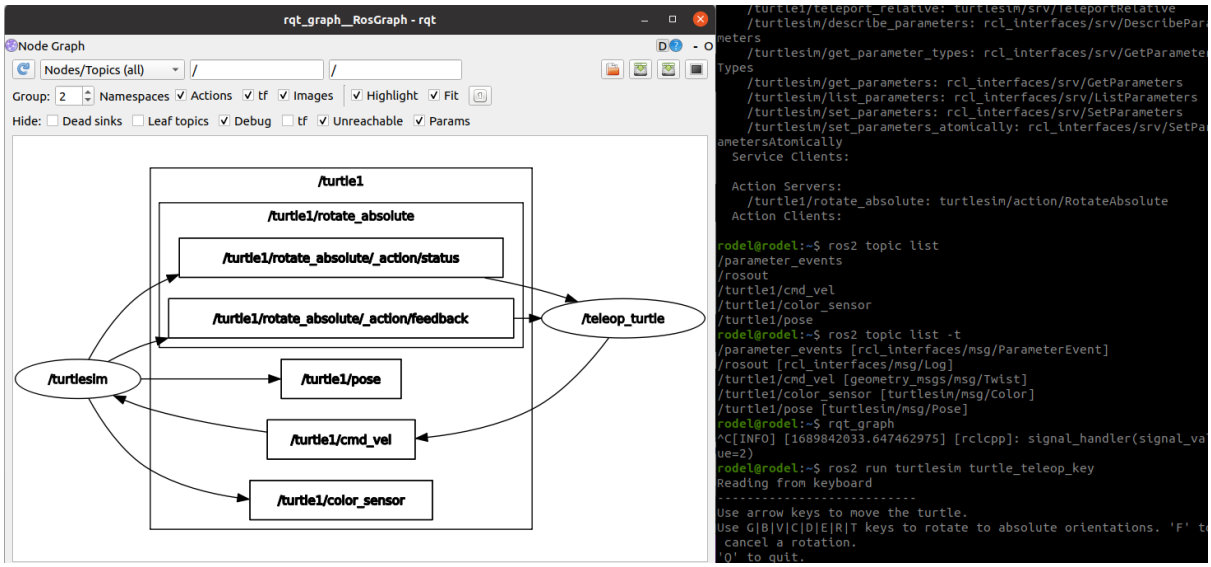
노드, 토픽, 서비스, 액션의 조회

```
~$ ros2 node list
/teleop_turtle
/turtlesim

~$ ros2 topic list
/parameter_events
/rosout
/turtle1/cmd_vel
/turtle1/color_sensor
/turtle1/pose

~$ ros2 service list
/clear
/kill
/reset
/spawn
/teleop_turtle/describe_parameters
/teleop_turtle/get_parameter_types
/teleop_turtle/get_parameters
/teleop_turtle/list_parameters
/teleop_turtle/set_parameters
/teleop_turtle/set_parameters_atomically
/turtle1/set_pen
/turtle1/teleport_absolute
/turtle1/teleport_relative
/turtlesim/describe_parameters
/turtlesim/get_parameter_types
/turtlesim/get_parameters
/turtlesim/list_parameters
/turtlesim/set_parameters
/turtlesim/set_parameters_atomically

~$ ros2 action list
/turtle1/rotate_absolute
```

turtlesim 노드 정보 확인 —> subscribe , publish 하는 메시지(토픽) 확인 가능

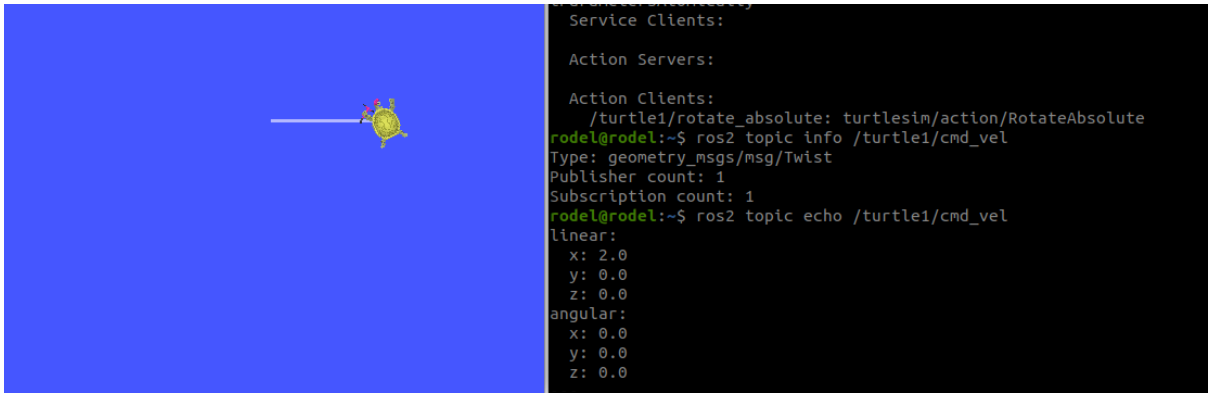
```
Subscribers:
  /parameter_events: rcl_interfaces/msg/ParameterEvent
  /turtle1/cmd_vel: geometry_msgs/msg/Twist
Publishers:
  /parameter_events: rcl_interfaces/msg/ParameterEvent
  /rosout: rcl_interfaces/msg/Log
  /turtle1/color_sensor: turtlesim/msg/Color
  /turtle1/pose: turtlesim/msg/Pose
Service Servers:
  /clear: std_srvs/srv/Empty
  /kill: turtlesim/srv/Kill
  /reset: std_srvs/srv/Empty
  /spawn: turtlesim/srv/Spawn
  /turtle1/set_pen: turtlesim/srv/SetPen
  /turtle1/teleport_absolute: turtlesim/srv/TeleportAbsolute
  /turtle1/teleport_relative: turtlesim/srv/TeleportRelative
  /turtlesim/describe_parameters: rcl_interfaces/srv/DescribeParameters
  /turtlesim/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
  /turtlesim/get_parameters: rcl_interfaces/srv/GetParameters
  /turtlesim/list_parameters: rcl_interfaces/srv/ListParameters
  /turtlesim/set_parameters: rcl_interfaces/srv/SetParameters
  /turtlesim/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
Service Clients:

Action Servers:
  /turtle1/rotate_absolute: turtlesim/action/RotateAbsolute
Action Clients:
```

rqt_graph :시각적으로 노드, 주고받는 토픽, publisher, subscriber을 알 수 있다.



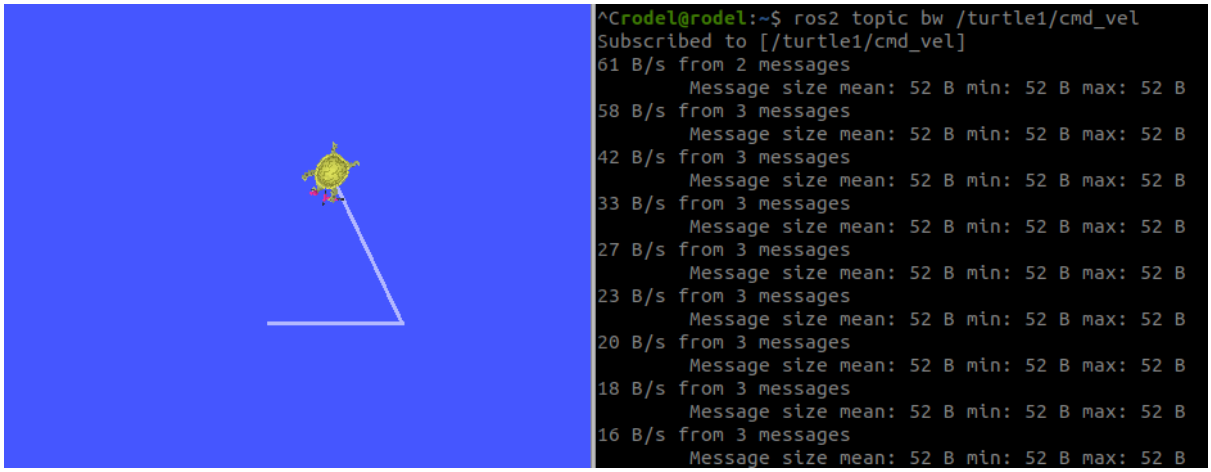
- node/Topic(all)을 해야 subscriber 없는 토픽들도 볼 수 있음.

토픽 정보 확인(ros2 topic info) + 토픽 내용 확인(ros2 topic echo)

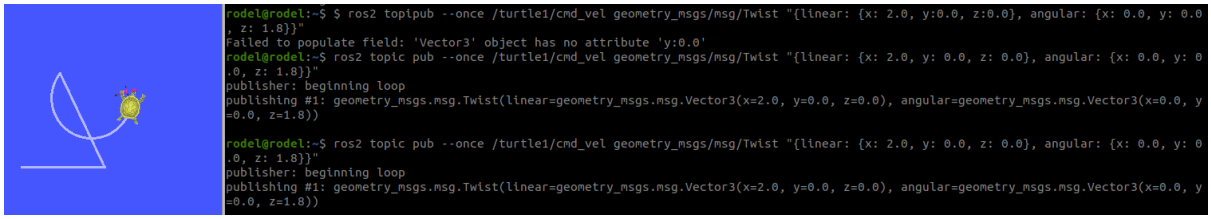


- turtle을 방향키로 움직였을 때 linear, angular 토픽 정보값이 나옴

토픽의대역폭(송수신 받는 메시지 크기) 확인 (ros2 topic bw)



토픽 퍼블리시(ros2 topic pub)



- —once 로 한번만 실행
- 노드 안의 퍼블리셔로 토픽 발행하지 않고 명령창에 직접 토픽 퍼블리시 가능

- 지속적인 실행 원할 경우 : —rate 1 추가(주기 1Hz) —> 주기적으로 거북이가 계속 회전한다

🤣 띄어쓰기 조심하자!

bag 기록 (ros2 bag record (토픽 이름))

```
~$ ros2 bag record /turtle1/cmd_vel
[INFO] [1689909083.819454402] [rosbag2_storage]: Opened database 'rosbag2_2023_07_21-12_11_23/rosbag2_2023_07_21-12_11_23_0.db3' for R
[INFO] [1689909083.830183653] [rosbag2_transport]: Listening for topics...
[INFO] [1689909083.830698810] [rosbag2_transport]: Subscribed to topic '/turtle1/cmd_vel'
[INFO] [1689909083.830785696] [rosbag2_transport]: All requested topics are subscribed. Stopping discovery...
```

- publish 되는 토픽을 파일 형태로 저장하고 필요할 때 다시 불러와 동일한 주기로 재생하는 기능
- 매번 로봇을 구동시켜 데이터 취득하기 힘들고 하더라도 로봇 상태값에 따라 결과값이 달라져서 불확실

→ 알고리즘 입력값 고정 후 반복 테스트하기 좋은 도구 : rosbag

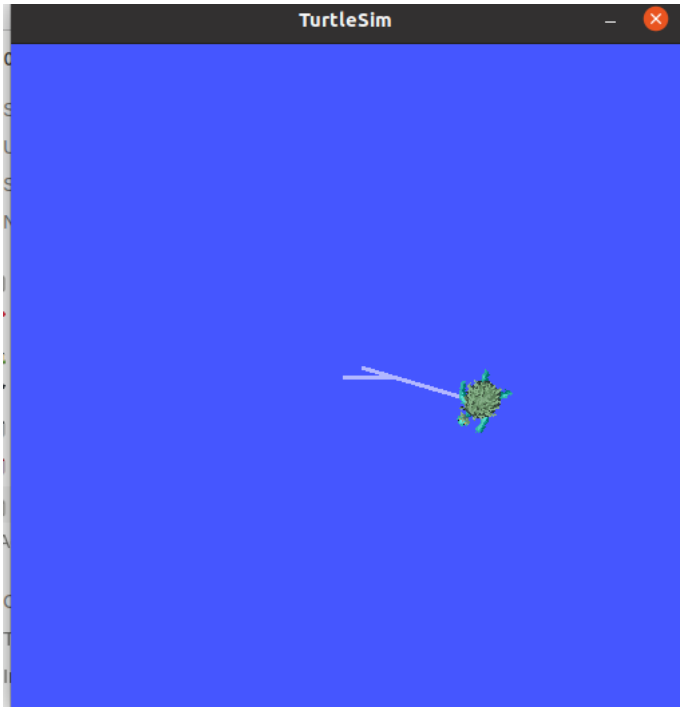
bag 정보(ros2 bag info(토픽 이름))

```
$ ros2 bag info rosbag2_2023_07_21-12_14_16

Files:             rosbag2_2023_07_21-12_14_16_0.db3
Bag size:          16.8 KiB
Storage id:        sqlite3
Duration:          7.24s
Start:             Jul 21 2023 12:14:20.668 (1689909260.668)
End:               Jul 21 2023 12:14:27.693 (1689909267.693)
Messages:          7
Topic information: Topic: /turtle1/cmd_vel | Type: geometry_msgs/msg/Twist | Count: 7 | Serialization Format: cdr
```

- 터틀봇을 7번 움직인 내용(Messages)이 기록되어있다

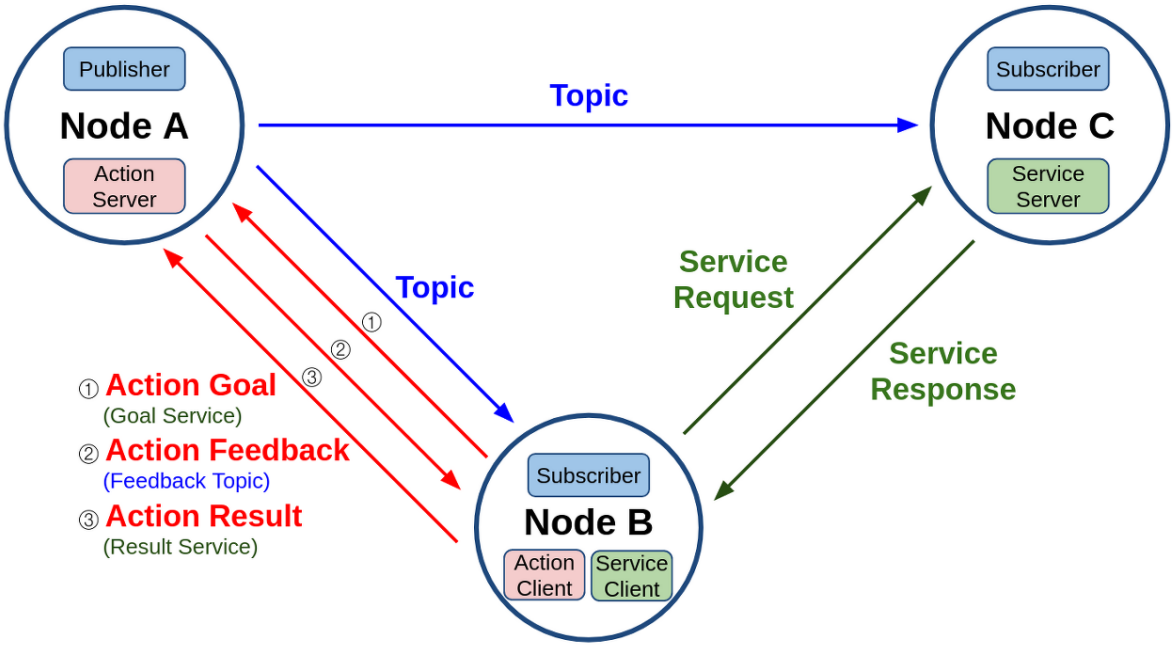
bag 재생(ros2 bag play(토픽 이름))

```
~$ ros2 bag play rosbag2_2023_07_21-12_14_16
[INFO] [1689909585.066727725] [rosbag2_storage]: Opened database 'rosbag2_2023_07_21-12_14_16/rosbag2_2023_07_21-12_14_16_0.db3' for R
```



- 기록했던 대로 잘 움직인다.

12. ROS2 서비스



- 동기식 양방향 메시지 송수신 방식
- Service Client : 서비스 요청(request) 하는 쪽
- Service Server : 서비스 응답(response) 하는 쪽

→서비스 응답은 요청한 쪽에게'만' 해준다.

- 즉 특정 요청을 하는 클라이언트 단과 요청받은 일을 수행한 후 결과값을 전달하는 서버 단과의 통신

서비스 목록확인(ros2 service list)

```
~$ ros2 service list -t
/clear [std_srvs/srv/Empty]
/kill [turtlesim/srv/Kill]
/reset [std_srvs/srv/Empty]
/spawn [turtlesim/srv/Spawn]
/teleop_turtle/describe_parameters [rcl_interfaces/srv/DescribeParameters]
/teleop_turtle/get_parameter_types [rcl_interfaces/srv/GetParameterTypes]
/teleop_turtle/get_parameters [rcl_interfaces/srv/GetParameters]
/teleop_turtle/list_parameters [rcl_interfaces/srv/ListParameters]
/teleop_turtle/set_parameters [rcl_interfaces/srv/SetParameters]
/teleop_turtle/set_parameters_atomically [rcl_interfaces/srv/SetParametersAtomically]
/turtle1/set_pen [turtlesim/srv/SetPen]
/turtle1/teleport_absolute [turtlesim/srv/TeleportAbsolute]
/turtle1/teleport_relative [turtlesim/srv/TeleportRelative]
/turtlesim/describe_parameters [rcl_interfaces/srv/DescribeParameters]
/turtlesim/get_parameter_types [rcl_interfaces/srv/GetParameterTypes]
/turtlesim/get_parameters [rcl_interfaces/srv/GetParameters]
/turtlesim/list_parameters [rcl_interfaces/srv/ListParameters]
/turtlesim/set_parameters [rcl_interfaces/srv/SetParameters]
/turtlesim/set_parameters_atomically [rcl_interfaces/srv/SetParametersAtomically]
```

- -t 를 붙여 데이터 타입과 함께 확인 가능

```
~$ ros2 service type /clear
std_srvs/srv/Empty
```

```
~$ ros2 service find std_srvs/srv/Empty
/clear
/reset
```

- 특정 서비스의 type을 묻거나, 반대로 특정 type갖는 서비스 조회

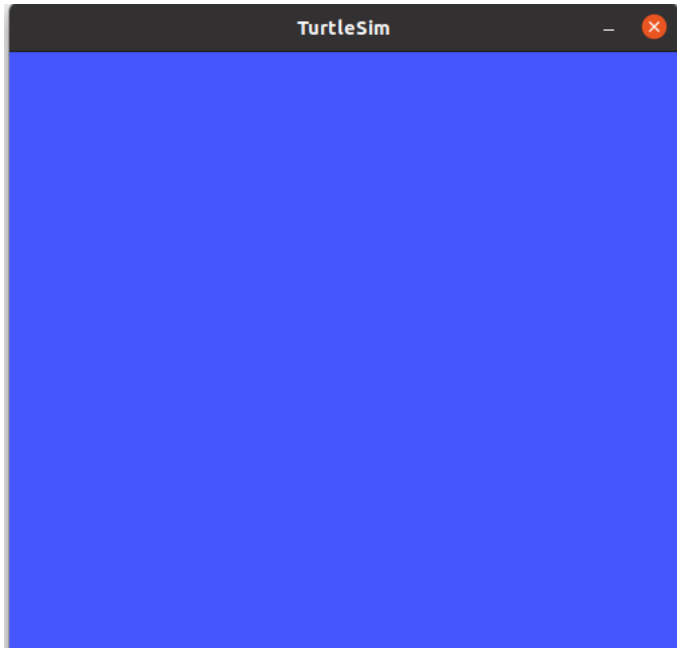
서비스 요청(ros2 service call)

```
ros2 service call <service_name> <service_type> "<arguments>"
```

```
$ ros2 run turtlesim turtle_teleop_key
-> 거북이를 움직여보자

~$ ros2 service call /clear std_srvs/srv/Empty
waiting for service to become available...
requester: making request: std_srvs.srv.Empty_Request()

response:
std_srvs.srv.Empty_Response()
```

- /clear 서비스를 요청해 거북이의 이동경로가 사라짐
- "<arguments>" 생략된 이유는 std_srvs/srv/Empty 라는 서비스 형태가 아무런 내용없는 형태로 사용가능하기 때문

```
~$ ros2 service call /kill turtlesim/srv/Kill "name: 'turtle1'"
waiting for service to become available...
requester: making request: turtlesim.srv.Kill_Request(name='turtle1')

response:
turtlesim.srv.Kill_Response()
```



- /kill 서비스로 거북이를 죽였다. 그는 좋은 거북이였다.

🔔 /kill turtlesim/srv/Kill 의 끝 K 대문자 조심

```
~$ ros2 service call /reset std_srvs/srv/Empty
waiting for service to become available...
requester: making request: std_srvs.srv.Empty_Request()

response:
std_srvs.srv.Empty_Response()
```

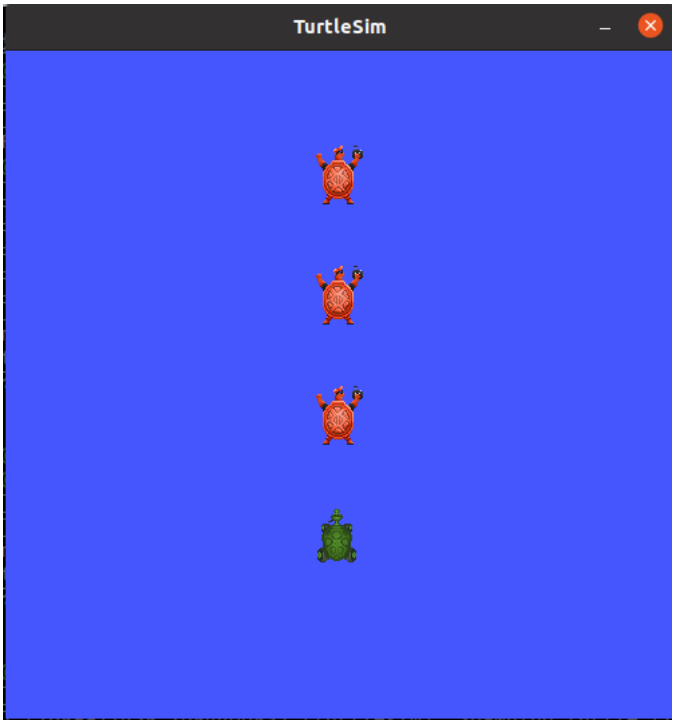
- /reset 으로 부활!

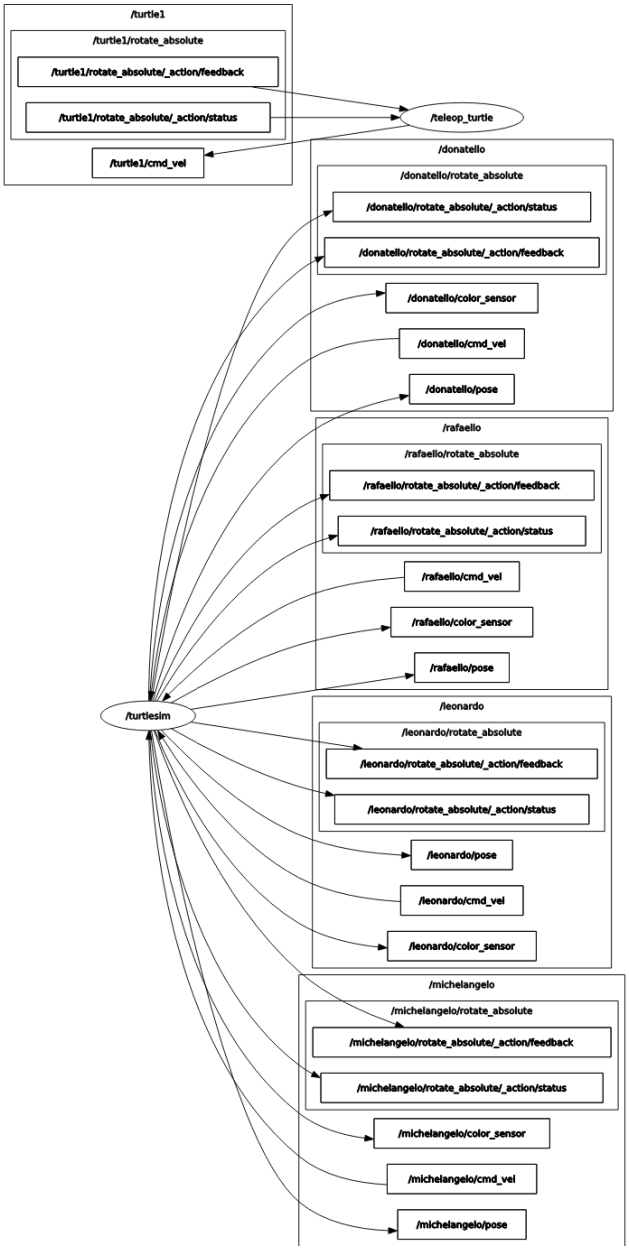
거북이 4마리

```
~$ ros2 service call /kill turtlesim/srv/Kill "name: 'turtle1'"
waiting for service to become available...
requester: making request: turtlesim.srv.Kill_Request(name='turtle1')
```

```
response:
turtlesim.srv.Kill_Response()

~$ ros2 service call /spawn turtlesim/srv/Spawn "{x: 5.5, y: 9, theta: 1.57, name: 'leonardo'}"
requester: making request: turtlesim.srv.Spawn_Request(x=5.5, y=9.0, theta=1.57, name='leonardo')

response:
turtlesim.srv.Spawn_Response(name='leonardo')

~$ ros2 service call /spawn turtlesim/srv/Spawn "{x: 5.5, y: 7, theta: 1.57, name: 'rafaello'}"
waiting for service to become available...
requester: making request: turtlesim.srv.Spawn_Request(x=5.5, y=7.0, theta=1.57, name='rafaello')

response:
turtlesim.srv.Spawn_Response(name='rafaello')~$ ros2 service call /spawn turtlesim/srv/Spawn "{x: 5.5, y: 5, theta: 1.57, name: 'miche
waiting for service to become available...
requester: making request: turtlesim.srv.Spawn_Request(x=5.5, y=5.0, theta=1.57, name='michelangelo')

response:
turtlesim.srv.Spawn_Response(name='michelangelo')

rodel@rodel:~$ ros2 service call /spawn turtlesim/srv/Spawn "{x: 5.5, y: 3, theta: 1.57, name: 'donatello'}"
requester: making request: turtlesim.srv.Spawn_Request(x=5.5, y=3.0, theta=1.57, name='donatello')

response:
turtlesim.srv.Spawn_Response(name='donatello')

~$ ros2 service call /spawn turtlesim/srv/Spawn "{x: 5.5, y: 5, theta: 1.57, name: 'michelangelo'}"
waiting for service to become available...
requester: making request: turtlesim.srv.Spawn_Request(x=5.5, y=5.0, theta=1.57, name='michelangelo')

response:
turtlesim.srv.Spawn_Response(name='michelangelo')

~$ ros2 service call /spawn turtlesim/srv/Spawn "{x: 5.5, y: 3, theta: 1.57, name: 'donatello'}"
requester: making request: turtlesim.srv.Spawn_Request(x=5.5, y=3.0, theta=1.57, name='donatello')

response:
turtlesim.srv.Spawn_Response(name='donatello')
```

```
~$ ros2 topic list
/donatello/cmd_vel
/donatello/color_sensor
/donatello/pose
/leonardo/cmd_vel
/leonardo/color_sensor
/leonardo/pose
/michelangelo/cmd_vel
/michelangelo/color_sensor
/michelangelo/pose
/parameter_events
/rafaello/cmd_vel
/rafaello/color_sensor
/rafaello/pose
/rosout
/turtle1/cmd_vel
```


TurtleSim


/turtle1
/turtle1/rotate_absolute
/turtle1/rotate_absolute/_action/feedback
/turtle1/rotate_absolute/_action/status
/turtle1/cmd_vel
/teleop_turtle
/donatello
/donatello/rotate_absolute
/donatello/rotate_absolute/_action/status
/donatello/rotate_absolute/_action/feedback
/donatello/color_sensor
/donatello/cmd_vel
/donatello/pose
/rafaello
/rafaello/rotate_absolute
/rafaello/rotate_absolute/_action/feedback
/rafaello/rotate_absolute/_action/status
/rafaello/cmd_vel
/rafaello/color_sensor
/rafaello/pose
/turtlesim
/leonardo
/leonardo/rotate_absolute
/leonardo/rotate_absolute/_action/feedback
/leonardo/rotate_absolute/_action/status
/leonardo/pose
/leonardo/cmd_vel
/leonardo/color_sensor
/michelangelo
/michelangelo/rotate_absolute
/michelangelo/rotate_absolute/_action/feedback
/michelangelo/rotate_absolute/_action/status
/michelangelo/color_sensor
/michelangelo/cmd_vel
/michelangelo/pose