

Week3

≡ 태그	
≡ 내용	

9장 패키지 설치와 노드 실행

turtlesim 이란?


turtlesim 은 ROS tutorials 에 있는 대표적인 예제 패키지 이다. turtlesim 패키지에는 다음의 Node 가 있다.

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

turtlesim 의 코드는 아래 github 를 참고한다.

ros_tutorials/turtlesim at humble · ros/ros_tutorials

Code used in tutorials found on ROS wiki. Contribute to ros/ros_tutorials development by creating an account on GitHub.

 https://github.com/ros/ros_tutorials/tree/humble/turtlesim

ros/ros_tutorials

Code used in tutorials found on ROS wiki

23

Contributors

9


Issues

663

Stars

515

Forks

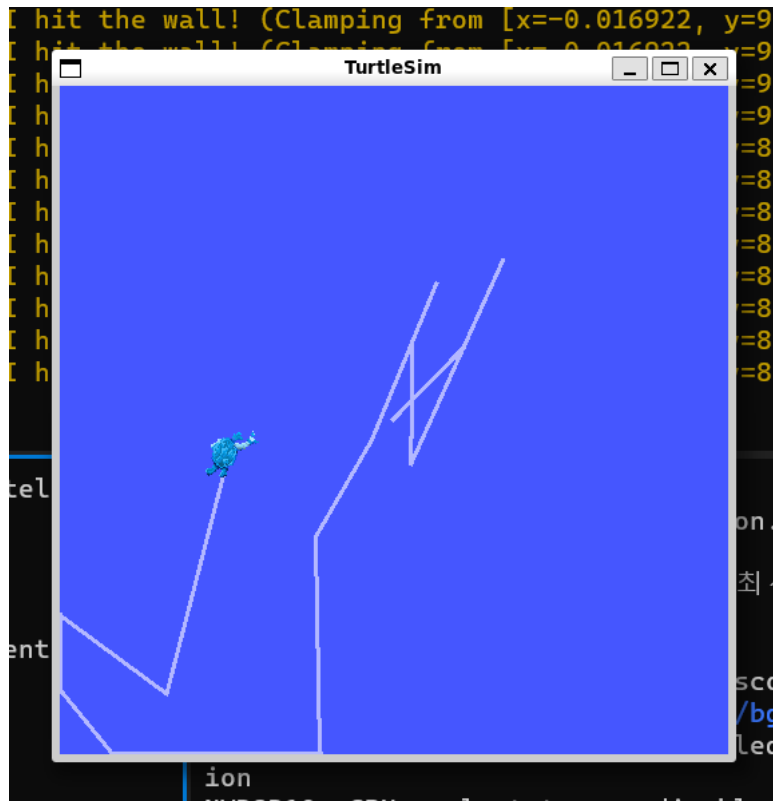


turtlesim 을 실행해보자!

명령줄 환경에서 turtlesim_node 노드와 turtle_teleop_key 를 각각 실행한다.

```
$ ros2 run turtlesim turtlesim_node
```

```
$ ros2 run turtlesim turtle_teleop_key
```



turtlesim 과 teleop_turtle 이 실행된 상태에서 node list 를 보면 아래와 같다.

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

topic list 명령어로 교환 가능한 topic 목록을 볼 수 있다.

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

service list 명령어로 교환 가능한 service 목록을 볼 수 있다.

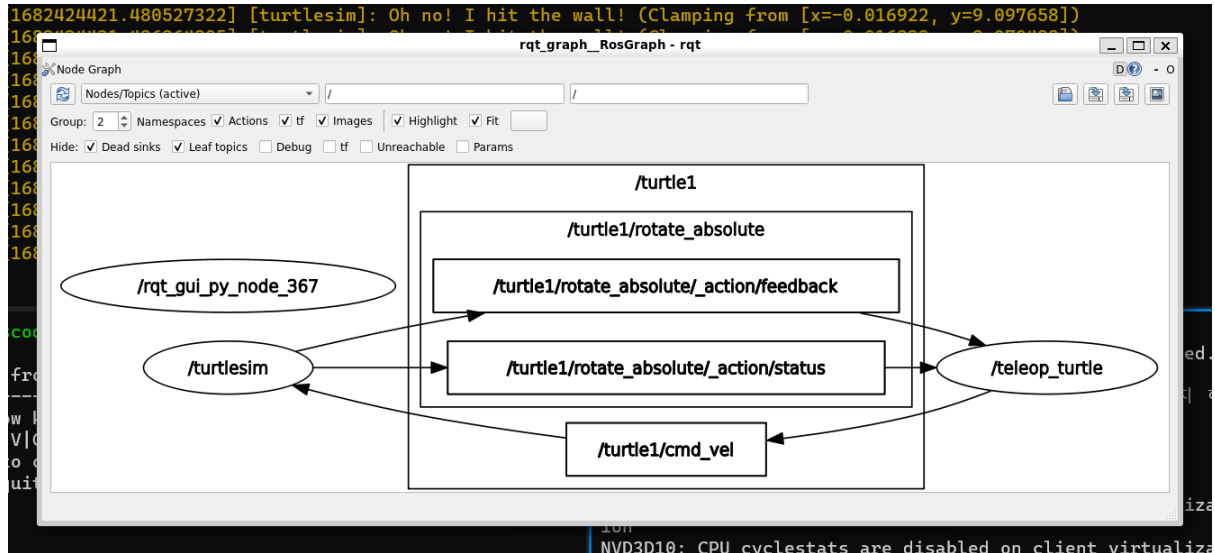
```
$ ros2 service list
/clear
/kill
/reset
/spawn
/teleop_turtle/describe_parameters
/teleop_turtle/get_parameter_types
/teleop_turtle/get_parameters
/teleop_turtle/list_parameters
/teleop_turtle/set_parameters
/teleop_turtle/set_parameters_atomically
/turtle1/set_pen
/turtle1/teleport_absolute
/turtle1/teleport_relative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

action list 명령어로 action 의 목록을 볼 수 있다.

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

rqt_graph 를 통한 노드와 토픽 시각화

ROS2 GUI 도구인 rqt_graph를 사용하면 노드와 토픽을 시각화할 수 있다.



10장 ROS2 노드와 데이터통신

노드와 데이터통신

- 노드는 ROS2 의 **최소실행단위** 로서, 하나의 ROS2 프로그램을 의미한다. (운영체제의 프로세스의 개념)
- 노드는 메시지를 통해서 서로 통신한다.
- 메시지의 형태는 Integer, Float, Bool, String 과 같이 다양하다.
- 메시지의 교환방법은 **토픽**, **서비스**, **액션**, **파라미터** 로 나눌 수 있다.

노드의 실행

명령줄 환경에서 ros2 run 명령 또는 launch 명령으로 노드를 실행할 수 있다.

```
$ ros2 run turtlesim turtlesim_node
$ ros2 run turtlesim turtlesim_teleop_key
```

실행중인 노드의 목록보기

명령줄 환경에서 ros2 node list 명령어를 통해 실행중인 노드의 목록을 볼 수 있다.

노드의 정보보기

명령줄 환경에서 node info 명령어를 통해 노드의 정보를 확인할 수 있다.

```
$ ros2 node info /turtlesim
/turtlesim
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:

```

11장 토픽

- 토픽은 비동기식 단방향 메시지 송수신 방식이다. 메시지를 송신하는 Publisher 와 메시지를 수신하는 Subscriber 간의 통신이다.

```

$ ros2 node info /turtlesim
/turtlesim
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

```

- turtlesim 노드는 Subscribers 로써, Twist 형식의 cmd_vel 를 수신한다.
- turtlesim 노드는 Publishers 로써, Pose 형식의 pose 를 발행한다.

토픽의 내용 확인 방법 (ros2 topic echo)

topic echo 명령어로 토픽의 값을 확인할 수 있다.

```

$ ros2 topic echo /turtle1/cmd_vel
linear:
  x: -2.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0토포토

```

토픽 저장하기

```

$ ros2 bag record /turtle1/cmd_vel
[INFO] [1682600953.746372298] [rosbag2_recorder]: Press SPACE for pausing/resuming
[INFO] [1682600953.748410390] [rosbag2_storage]: Opened database 'rosbag2_2023_04_27-22_09_13/rosbag2_2023_04_27-22_09_13_0.db3' for R
[INFO] [1682600953.748501538] [rosbag2_recorder]: Event publisher thread: Starting
[INFO] [1682600953.750114295] [rosbag2_recorder]: Listening for topics...
[INFO] [1682600962.697019041] [rclcpp]: signal_handler(signum=2)
[INFO] [1682600962.780164254] [rosbag2_recorder]: Event publisher thread: Exiting

```

12장 서비스

- 서비스는 동기식 양방향 메시지 송수신 방식이다. 요청하는 쪽을 서비스 클라이언트, 요청받아 응답하는 쪽을 서비스 서버라고 한다.
- 명령줄 라인에서 service list 로 서비스의 목록을 볼 수 있다.

```

$ ros2 service list
/clear
/kill
/reset
/spawn

```

```
/turtle1/set_pen  
/turtle1/teleport_absolute  
/turtle1/teleport_relative
```

- 예를 들어 kill 서비스는 turtlesim/srv/Kill 형태임을 확인할 수 있다.

```
$ ros2 service type /kill  
turtlesim/srv/Kill
```

- 명령줄 라인에서 find 명령어로 특정 서비스를 찾을 수 있다.

```
$ ros2 service find turtlesim/srv/Kill  
/kill
```

서비스 요청

- 명령줄 라인에서 service call 로 서비스를 요청할 수 있다.
- 예를 들어 /clear 서비스를 요청하면 화면이 완전히 지워진다.
- 예를 들어 /kill 서비스를 요청하면 거북이가 사라진다.