

ROS 2

[06 – ROS2 PKG Overview]

한국폴리텍대학교 성남캠퍼스



Homework Review



Homework Review

❖ Missions

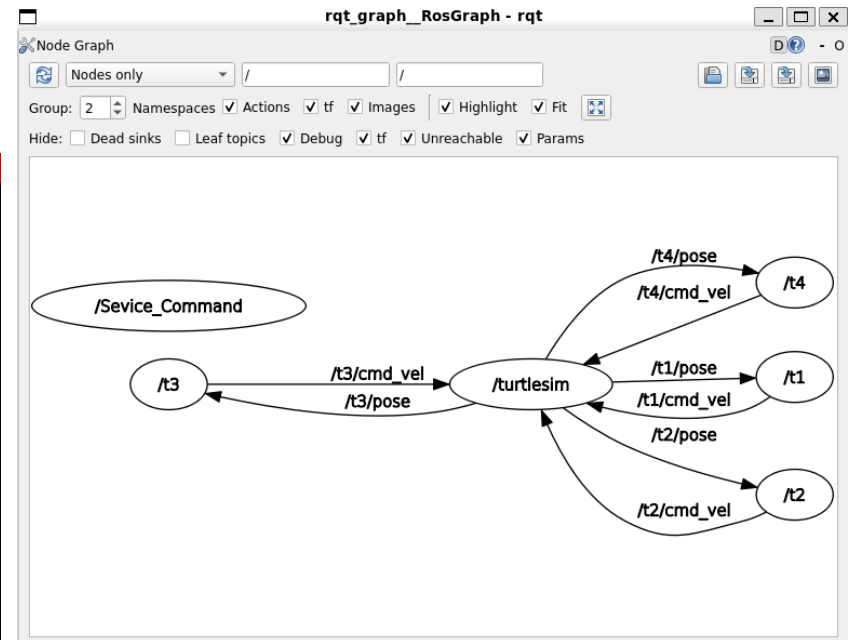
1. 정해진 구역 내에서 랜덤하게 주행하는 거북이를 클래스로 설계하기
2. Turtlesim을 Service Server로 대상을 정하고 5가지 이상의 서비스를 요청할 수 있는 Client를 클래스로 설계하기

Homework Review



```
8. exit
Select num = 6
Turtle name = t1
X Y Theta = 3 3 0
1. reset
2. clear
3. spawn
4. kill
5. set pen
6. teleport abs
7. teleport rel
8. exit
Select num =
```

```
r
Remove Turtle name :t3
a
node name = t2
Turtle name = t2
Low limit = 3
High limit = 7
a
node name = t3
Turtle name = t3
Low limit = 5
High limit = 9
```



1 Package Overview

1

Package Overview

❖ 패키지 작업을 위한 워크스페이스

```
$ mkdir -p ~/ros2_work/src
```

- mkdir -p 는 ros2_work 폴더 아래에 src라는 폴더까지 한번에 만들기 위해 사용
- ros2_work 폴더는 패키지 작업을 위한 기본 공간
- src폴더는 패키지 작업을 위한 소스코드가 저장되는 공간
- 워크스페이스는 여러 곳에 위치해도 상관없음

❖ 패키지 작업을 위한 워크스페이스

```
~/ros2_work$ colcon build
```

```
daesung@DSThinkPad:~/ros2_work$ ls
src
daesung@DSThinkPad:~/ros2_work$ colcon build

Summary: 0 packages finished [0.18s]
daesung@DSThinkPad:~/ros2_work$ ls
build  install  log  src
daesung@DSThinkPad:~/ros2_work$ |
```

- colcon build 명령은 src폴더에 있는 소스코드들을 빌드하는 명령
- 현재 소스코드가 없기 때문에 작업한 내용이 없음을 확인할 수 있음
- 빌드 명령 이후 파일을 조회해 보면 build, install, log 폴더가 추가로 생성되는 것을 알 수 있음

❖ 패키지 작업을 위한 워크스페이스

- Build
 - ✓ 빌드 설정파일 저장
- Install
 - ✓ 패키지 내에서 생성한 msg, srv, action 과 관련된 헤더파일 또는 모듈, 패키지 라이브러리, 실행파일 저장
- Log
 - ✓ 빌드할 때 생성된 내용들 저장
- Src
 - ✓ 사용자가 패키지를 만들기 위해 소스코드를 작성하여 저장

❖ 패키지 만들기 무작정 따라하기

```
~/ros2_work/src$ ros2 pkg create --build-type ament_python --  
node-name first_node first_package
```

- ros2 pkg create 명령을 이용해서 패키지 생성
- --build-type ament_python 옵션을 이용해서 빌드 타입 지정
 - ✓ Ros2에서 사용하는 빌드 도구는 ament
 - ✓ Python을 사용할 경우 ament-python으로 지정
- --node-name 옵션을 이용해서 노드 관련 정보 설정
 - 패키지를 생성할 때 노드도 함께 생성해 달라는 옵션
 - 해당 옵션은 패키지에 노드를 등록하는 작업이 어렵지 않아 자주 쓰지 않음

```
ros2 pkg create --build-type ament_python <package_name>
```

❖ 패키지 만들기 무작정 따라하기

```
~/ros2_work/src$ ros2 pkg create --build-type ament_python --node-name first_node first_package
```

```
daesung@DSThinkPad:~/ros2_work/src$ ros2 pkg create --build-type ament_python --node-name first_node first_package
going to create a new package
package name: first_package
destination directory: /home/daesung/ros2_work/src
package format: 3
version: 0.0.0
description: TODO: Package description
maintainer: ['daesung <daesung7723@gmail.com>']
licenses: ['TODO: License declaration']
build type: ament_python
dependencies: []
node_name: first_node
creating folder ./first_package
creating ./first_package/package.xml
creating source folder
creating folder ./first_package/first_package
creating ./first_package/setup.py
creating ./first_package/setup.cfg
creating folder ./first_package/resource
creating ./first_package/resource/first_package
creating ./first_package/first_package/__init__.py
creating folder ./first_package/test
creating ./first_package/test/test_copyright.py
creating ./first_package/test/test_flake8.py
creating ./first_package/test/test_pep257.py
creating ./first_package/first_package/first_node.py

[WARNING]: Unknown license 'TODO: License declaration'. This has been set in the package.xml, but no LICENSE file has been created.
It is recommended to use one of the ament license identifiers:
Apache-2.0
BSL-1.0
BSD-2.0
BSD-2-Clause
BSD-3-Clause
GPL-3.0-only
LGPL-3.0-only
MIT
MIT-0
```

❖ 패키지 만들기 무작정 따라하기

- 폴더 구조를 파악할 수 있는 도구인 tree 설치

Sudo apt install tree

- 설치 후 구조 확인

```
daesung@DSThinkPad:~/ros2_work/src$ tree
```

```
.
├── first_package
│   ├── first_package
│   │   ├── first_node.py
│   │   └── __init__.py
│   ├── package.xml
│   ├── resource
│   │   └── first_package
│   ├── setup.cfg
│   ├── setup.py
│   └── test
│       ├── test_copyright.py
│       ├── test_flake8.py
│       └── test_pep257.py
```

```
4 directories, 9 files
```

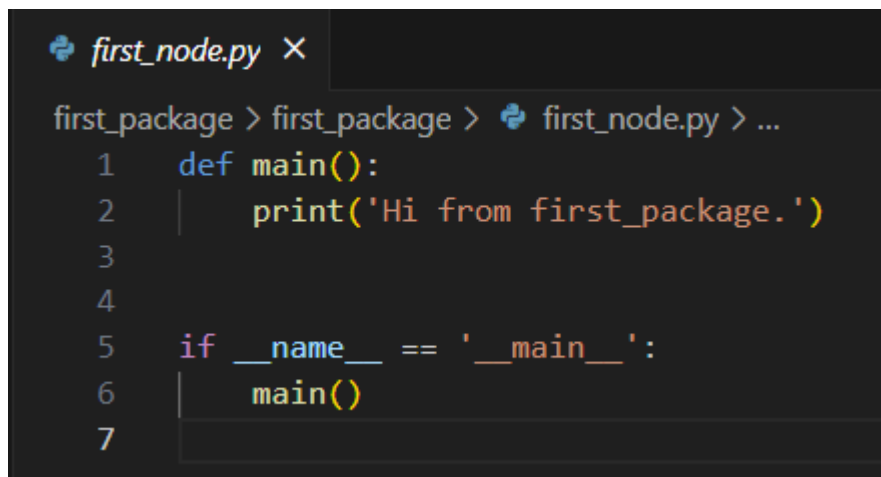
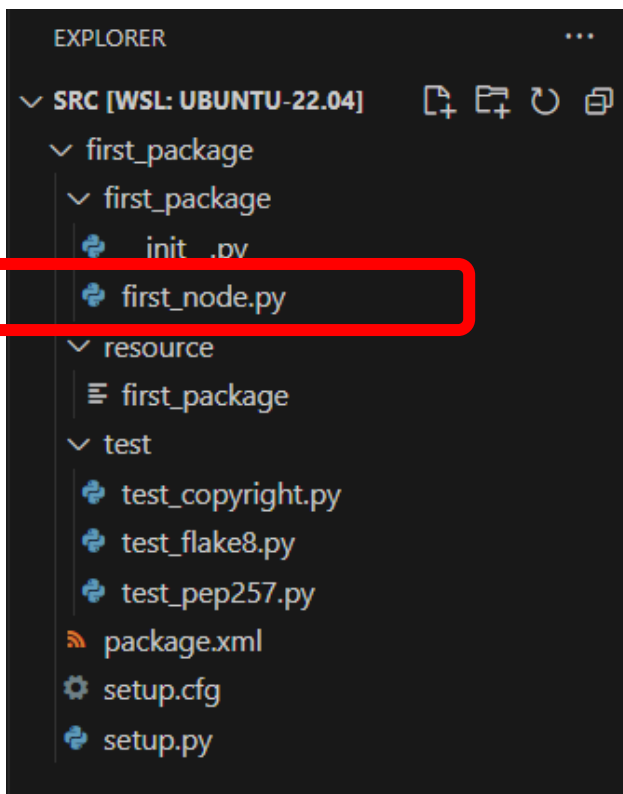
code .

- 현재 폴더의 모든 파일과
폴더를 여는 명령

❖ 패키지 만들기 무작정 따라하기

- 현재 폴더의 모든 파일과 폴더를 VS Code에서 열기

code .



❖ 패키지 만들기 무작정 따라하기

- Src 폴더의 상위 폴더인 워크스페이스 폴더로 이동하여 패키지 빌드

colcon build

```
daesung@DSThinkPad:~/ros2_work$ colcon build
Starting >>> first_package
Finished <<< first_package [0.60s]

Summary: 1 package finished [0.96s]
daesung@DSThinkPad:~/ros2_work$ |
```

- 항상 워크스페이스 폴더에서 colcon으로 빌드
- 상기 명령어는 별다른 옵션 없이 워크스페이스 내의 모든 패키지를 빌드
- 옵션을 통해 특정 패키지만 빌드할 수 있음

❖ 패키지 만들기 무작정 따라하기

- 현재 상태는 빌드까지 진행된 상태이며 실행이 된 상태는 아님

```
daesung@DSThinkPad:~/ros2_work$ ros2 run first_package first_node
Package 'first_package' not found
daesung@DSThinkPad:~/ros2_work$ |
```

- Install 폴더의 local_setup.bash를 불러오는 작업을 먼저 해야 함

```
daesung@DSThinkPad:~/ros2_work$ ls
build  install  log  src
daesung@DSThinkPad:~/ros2_work$ cd install/
daesung@DSThinkPad:~/ros2_work/install$ ls
COLCON_IGNORE      local_setup.sh          setup.bash
first_package      _local_setup_util_ps1.py  setup.ps1
local_setup.bash   _local_setup_util_sh.py  setup.sh
local_setup.ps1    local_setup.zsh          setup.zsh
daesung@DSThinkPad:~/ros2_work/install$ source local_setup.bash
daesung@DSThinkPad:~/ros2_work/install$ ros2 run first_package first_node
Hi from first_package.
daesung@DSThinkPad:~/ros2_work/install$ |
```

❖ 패키지 만들기 무작정 따라하기

- Ros2 실행을 위해 ~/.bashrc에 alias로 추가한 것과 같이 본인의 워크스페이스 패키지 실행을 위해 추가

```
124 alias rlb="source ~/.bashrc; echo \"bash is reloaded!\""  
125 alias pyrun="/bin/python3.10"  
126 alias mywk="source ~/ros2_work/install/local_setup.bash"
```

```
daesung@DSThinkPad:~$ humble  
ROS2 Humble is activated!  
ROS_Domain is 23  
daesung@DSThinkPad:~$ ros2 run first_package first_node  
Package 'first_package' not found  
daesung@DSThinkPad:~$ mywk  
daesung@DSThinkPad:~$ ros2 run first_package first_node  
Hi from first_package.  
daesung@DSThinkPad:~$ |
```

2 Package with Topic

❖ Subscriber 추가

- VS Code 실행 후
- Src/first_package/first_package 폴더에 topic_subscriber.py 추가

```
daesung@DSThinkPad:~/ros2_work/src$ tree
.
├── first_package
│   ├── first_package
│   │   ├── first_node.py
│   │   ├── __init__.py
│   │   └── topic_subscriber.py
│   ├── package.xml
│   ├── resource
│   │   └── first_package
│   ├── setup.cfg
│   ├── setup.py
│   └── test
│       ├── test_copyright.py
│       ├── test_flake8.py
│       └── test_pep257.py
4 directories, 10 files
```

❖ Subscriber 추가

- topic_subscriber.py 코드

```
import rclpy as rp
from rclpy.node import Node
from turtlesim.msg import Pose

class TurtlesimSubscriber(Node):
    def __init__(self):
        super().__init__('Turtlesim_Sub_node')
        self.sub = self.create_subscription(Pose, '/turtle1/pose', self.callback, 10)

    def callback(self, msg):
        print('X : %.2f Y: %.2f'%(msg.x, msg.y))

def main(args=None):
    rp.init(args=args)

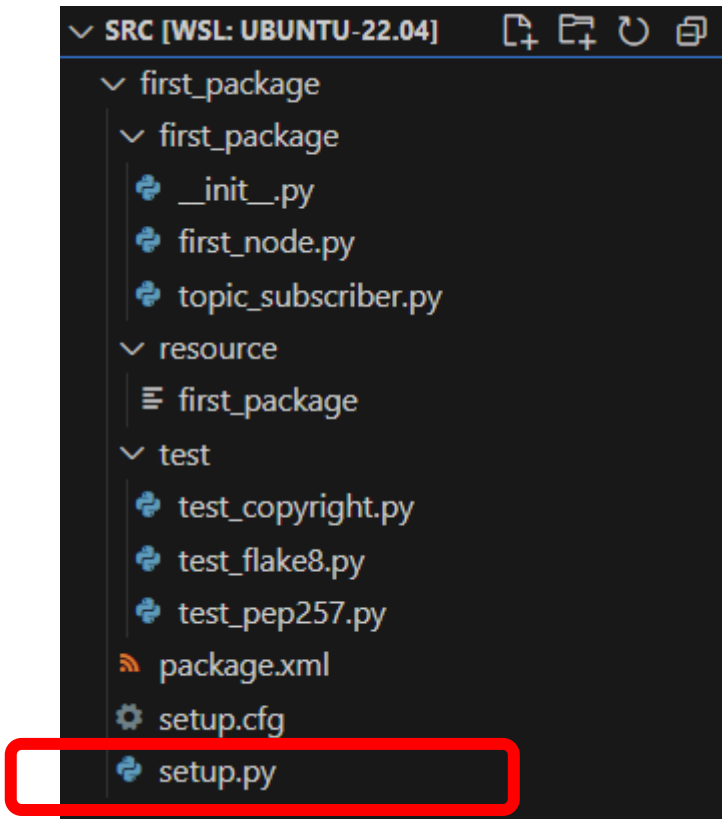
    turtlesim_sub = TurtlesimSubscriber()
    rp.spin(turtlesim_sub)

    turtlesim_sub.destroy_node()
    rp.shutdown()

if __name__ == '__main__':
    main()
```

❖ Subscriber 추가

- 추가한 노드 실행을 위해서는 setup.py 수정 필요



❖ Subscriber 추가

- 추가한 노드 실행을 위해서는 setup.py 수정 필요

```
entry_points={
    'console_scripts': [
        'first_node = first_package.first_node:main',
        'topic_subscriber = first_package.topic_subscriber:main'
    ],
},
```

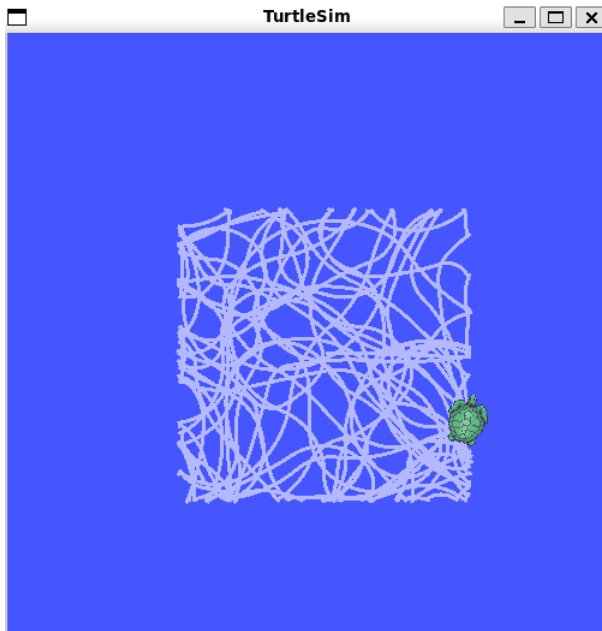
- 새로 추가한 topic_subscriber파일의 main함수 등록
- 수정된 내용이 반영 될 수 있도록 다시 빌드

```
daesung@DSThinkPad:~/ros2_work$ colcon build
Starting >>> first_package
Finished <<< first_package [0.63s]

Summary: 1 package finished [0.82s]
daesung@DSThinkPad:~/ros2_work$
```

❖ Subscriber 추가

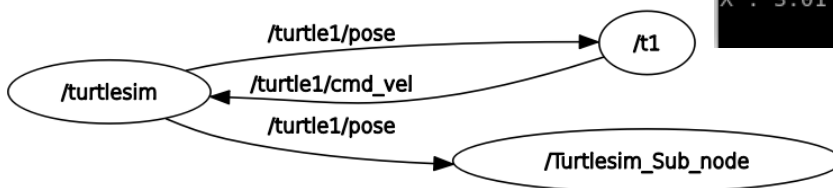
- Turtlesim_node를 실행한 후 내가 만든 패키지의 subscriber 노드 실행



```
daesung@DSThinkPad: ~/ros2_work
daesung@DSThinkPad: ~ 74x11
ROS2 Humble is activated!
ROS_Domain is 23
daesung@DSThinkPad:~/ros2_work/src$ cd ~
daesung@DSThinkPad:~$ ros2 run turtlesim turtlesim_node
QStandardPaths: wrong permissions on runtime directory /run/user/1000/, 07
55 instead of 0700
[INFO] [1727794893.024745090] [turtlesim]: Starting turtlesim with node na
me /turtlesim
[INFO] [1727794893.029869388] [turtlesim]: Spawning turtle [turtle1] at x=
[5.544445], y=[5.544445], theta=[0.000000]

daesung@DSThinkPad: ~/ros2_work 36x11
X : 3.17 Y: 4.94
X : 3.15 Y: 4.90
X : 3.13 Y: 4.86
X : 3.12 Y: 4.81
X : 3.10 Y: 4.77
X : 3.08 Y: 4.73
X : 3.07 Y: 4.68
X : 3.05 Y: 4.64
X : 3.03 Y: 4.60
X : 3.01 Y: 4.56

daesung@DSThinkPad: ~/test 36x11
daesung@DSThinkPad:~$ ls
ROS2 ros2_work test
daesung@DSThinkPad:~$ cd test/
daesung@DSThinkPad:~/test$ pyrun 04_
02_class_autonomous_moving.py
a
node name = t1
Turtle name = turtle1
Low limit = 3
High limit = 8
```



❖ Publisher 추가

- topic_publisher.py 코드 추가 → setup.py 수정 → 빌드 → Publisher 실행

```
import rclpy as rp
from rclpy.node import Node
from geometry_msgs.msg import Twist

class TurtlesimPublisher(Node):
    def __init__(self):
        super().__init__('turtlesim_pub_node')
        self.pub = self.create_publisher(Twist, '/turtle1/cmd_vel', 10)
        self.timer = self.create_timer(0.5, self.timer_callback)

    def timer_callback(self):
        msg = Twist()
        msg.linear.x = 2.0
        msg.angular.z = 2.0
        self.pub.publish(msg)

def main(args=None):
    rp.init(args=args)

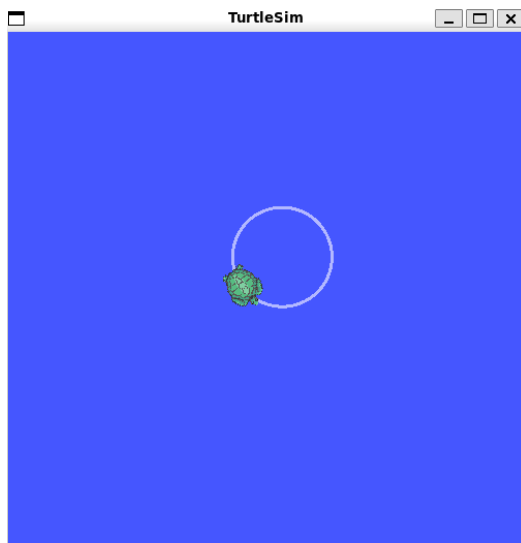
    turtlesim_pub = TurtlesimPublisher()
    rp.spin(turtlesim_pub)

    turtlesim_pub.destroy_node()
    rp.shutdown()

if __name__ == '__main__':
    main()
```

❖ Publisher 추가

- topic_publisher.py 코드 추가 → setup.py 수정 → 빌드 → Publisher 실행

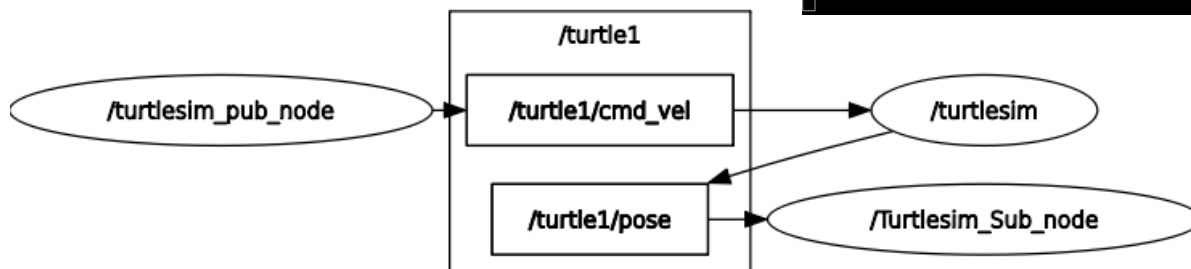


```
daesung@DSThinkPad: ~/test
[INFO] [1727796635.355644350] [turtle1]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]
[WARN] [1727796635.455995212] [turtle1]: failed to send response to /reset (timeout): client will not receive response, at ./src/rmw_response.cpp:154, at ./src/rcl/service.c:314

daesung@DSThinkPad: ~/test 36x11
daesung@DSThinkPad:~/test$ ros2 run first_package topic_publisher

daesung@DSThinkPad: ~/ros2_work 36x11
X : 4.89 Y: 7.32
X : 4.87 Y: 7.30
X : 4.84 Y: 7.27
X : 4.82 Y: 7.25
X : 4.80 Y: 7.23
X : 4.78 Y: 7.20
X : 4.76 Y: 7.18
X : 4.74 Y: 7.16
X : 4.72 Y: 7.13
X : 4.70 Y: 7.10

daesung@DSThinkPad: ~/test 36x11
daesung@DSThinkPad:~/test$ rqt_graph
QStandardPaths: wrong permissions on runtime directory /run/user/1000/, 0755 instead of 0700
```



2

Package with Topic

❖ Mission

- 패키지에 지정된 구간을 자율주행 할 수 있는 Topic을 Publishing하는 노드를 추가하여 `ros2 run first_package <file_name>`으로 실행 하기

3

Message

❖ Message 정의를 위한 패키지 만들기

- Message를 직접 정의하고 사용하는 방법에 대해 강의
- 메시지 타입 생성을 위해 다음과 같은 패키지 만들기

```
~/ros2_work/src$ ros2 pkg create --build-type ament_cmake  
first_package_msg
```

```
daesung@DSThinkPad:~/ros2_work/src$ tree  
.  
├── first_package  
│   ├── first_node.py  
│   ├── __init__.py  
│   ├── topic_publisher.py  
│   └── topic_subscriber.py  
├── package.xml  
├── resource  
│   └── first_package  
├── setup.cfg  
├── setup.py  
├── test  
│   ├── test_copyright.py  
│   ├── test_flake8.py  
│   └── test_pep257.py  
└── first_package_msg  
    ├── CMakeLists.txt  
    ├── include  
    │   └── first_package_msg  
    ├── package.xml  
    └── src  
8 directories, 13 files
```

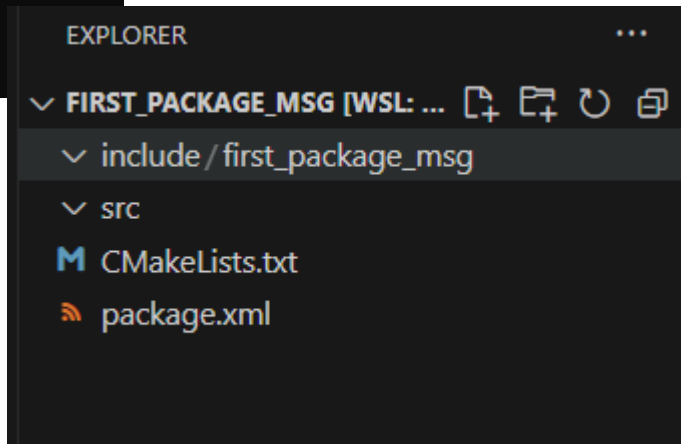
```
└── first_package_msg  
    ├── CMakeLists.txt  
    ├── include  
    │   └── first_package_msg  
    ├── package.xml  
    └── src
```

❖ Message 정의를 위한 패키지 수정

- 해당 패키지 폴더로 이동하고 VS Code를 실행하여 폴더 전체 열기

```
daesung@DSThinkPad:~/ros2_work/src$ ls
first_package  first_package_msg
daesung@DSThinkPad:~/ros2_work/src$ cd first_package_msg/
daesung@DSThinkPad:~/ros2_work/src/first_package_msg$ code .
daesung@DSThinkPad:~/ros2_work/src/first_package_msg$ tree
.
├── CMakeLists.txt
├── include
│   └── first_package_msg
├── package.xml
└── src

3 directories, 2 files
daesung@DSThinkPad:~/ros2_work/src/first_package_msg$ |
```



❖ Message 정의를 위한 패키지 수정

- Msg 폴더를 생성하고 ~~.msg 파일 생성

```
daesung@DSThinkPad:~/ros2_work/src/first_package_msg$ tree
.
├── CMakeLists.txt
├── include
│   └── first_package_msg
├── msg
│   └── FirstMsg.msg
├── package.xml
└── src

4 directories, 3 files
daesung@DSThinkPad:~/ros2_work/src/first_package_msg$ |
```

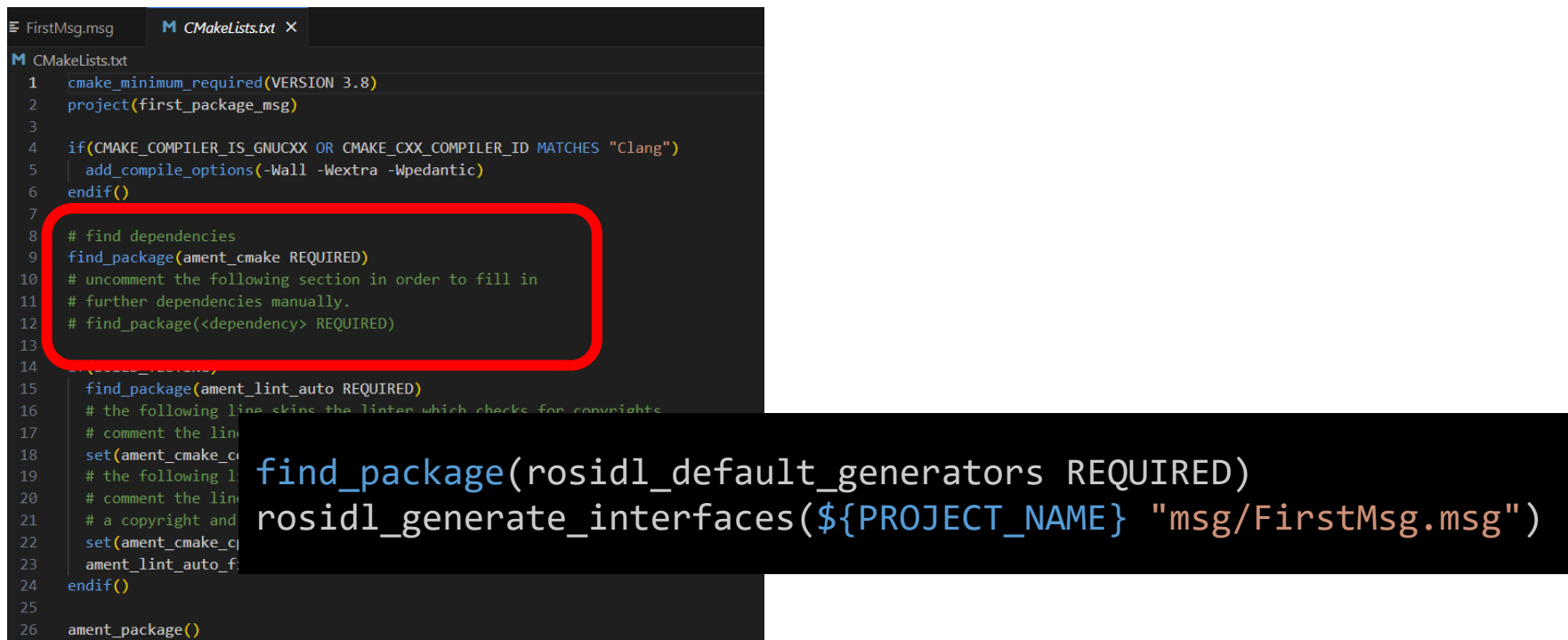
≡ FirstMsg.msg X

src > ≡ FirstMsg.msg

```
1 float32 cmd_vel_linear
2 float32 cmd_vel_angular
3
4 float32 pose_x
5 float32 pose_y
6 float32 linear_vel
7 float32 angular_vel
```

❖ Message 정의를 위한 패키지 빌드

- 새롭게 정의한 message를 사용하기 위해서는 두 개의 파일을 수정해야 함
 - ✓ CMakeLists.txt
 - ✓ package.xml
- 먼저 CMakeLists.txt 파일의 다음 부분 마지막에 코드 추가



```
1 cmake_minimum_required(VERSION 3.8)
2 project(first_package_msg)
3
4 if(CMAKE_COMPILER_IS_GNUCXX OR CMAKE_CXX_COMPILER_ID MATCHES "Clang")
5   add_compile_options(-Wall -Wextra -Wpedantic)
6 endif()
7
8 # find dependencies
9 find_package(ament_cmake REQUIRED)
10 # uncomment the following section in order to fill in
11 # further dependencies manually.
12 # find_package(<dependency> REQUIRED)
13
14 # find_package(ament_lint_auto REQUIRED)
15 # the following line skips the linter which checks for copyrights
16 # comment the line below to enable the linter
17 # set(ament_cmake_cpplint ament_lint_auto_cpplint)
18 # the following line skips the linter which checks for copyrights
19 # comment the line below to enable the linter
20 # a copyright and license can be included in this section
21 # find_package(ament_lint_auto REQUIRED)
22 # the following line skips the linter which checks for copyrights
23 # comment the line below to enable the linter
24 # set(ament_cmake_cpplint ament_lint_auto_cpplint)
25
26 find_package(ament_cmake REQUIRED)
27 find_package(rosidl_default_generators REQUIRED)
28 rosidl_generate_interfaces(${PROJECT_NAME} "msg/FirstMsg.msg")
29
30 ament_package()
```

❖ Message 정의를 위한 패키지 빌드

- package.xml 파일의 <export> 위에 다음과 같은 코드 추가

```
package.xml
1  <?xml version="1.0"?>
2  <?xml-model href="http://download.ros.org/schema/package_format3.xsd" schematypens="http://www.w3.org/2001/XMLSchema"?>
3  <package format="3">
4    <name>first_package_msg</name>
5    <version>0.0.0</version>
6    <description>TODO: Package description</description>
7    <maintainer email="daesung7723@gmail.com">daesung</maintainer>
8    <license>TODO: License declaration</license>
9
10   <buildtool_depend>ament_cmake</buildtool_depend>
11
12   <test_depend>ament_lint_auto</test_depend>
13   <test_depend>ament_lint_common</test_depend>
14
15   <export>
16     <buildtool_depend>ament_cmake</buildtool_depend>
17   </export>
18 </package>
```

```
<build_depend>roscpp</build_depend>
<exec_depend>roscpp</exec_depend>
<member_of_group>roscpp</member_of_group>
```

❖ Message 정의를 위한 패키지 빌드

- 두 개의 파일을 모두 수정한 후 워크스페이스 폴더로 이동하여 colcon build 실행

```
daesung@DSThinkPad:~/ros2_work$ colcon build
Starting >>> first_package
Starting >>> first_package_msg
Finished <<< first_package [0.66s]
Finished <<< first_package_msg [1.50s]

Summary: 2 packages finished [1.91s]
daesung@DSThinkPad:~/ros2_work$ |
```

- Ros2 interface show를 이용하여 작성된 내용 확인

```
daesung@DSThinkPad:~$ ros2 interface show first_package_msg/msg/FirstMsg
float32 cmd_vel_linear
float32 cmd_vel_angular

float32 pose_x
float32 pose_y
float32 linear_vel
float32 angular_vel
daesung@DSThinkPad:~$ |
```

❖ 정의된 Message 활용

- 현재 사용자 정의 msg는 아래와 같이 turtlesim이 발행하는 pose 토픽, cmd_vel 토픽을 구독할 수 있음

```
daesung@DSThinkPad:~$ ros2 interface show first_package_msg/msg/FirstMsg
float32 cmd_vel_linear
float32 cmd_vel_angular

float32 pose_x
float32 pose_y
float32 linear_vel
float32 angular_vel
daesung@DSThinkPad:~$ |
```

- 이를 저장하고 발행하는 패키지를 작성

❖ 정의된 Message 활용

- 처음 만들어서 작성한 first_package를 활용
- 테스트를 위한 ~~.py 파일 만들기

```
import rclpy as rp
from rclpy.node import Node
from turtlesim.msg import Pose

class CmdPose(Node):
    def __init__(self):
        super().__init__('turtle_cmd_pose_node')
        self.sub_pose = self.create_subscription(Pose, '/turtle1/pose', self.callback_pose, 10)

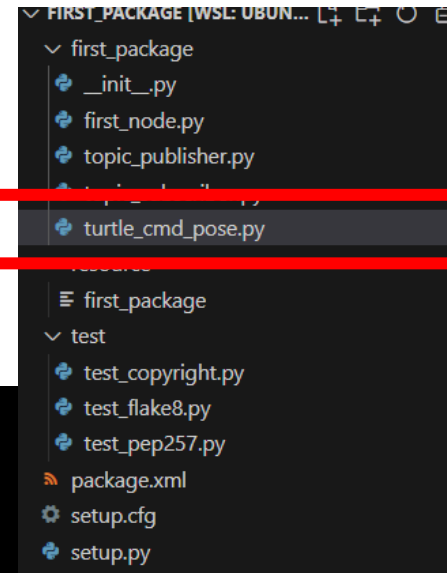
    def callback_pose(self, msg):
        print(msg)

def main(args=None):
    rp.init(args=args)

    turtle_cmd_pose = CmdPose()
    rp.spin(turtle_cmd_pose)

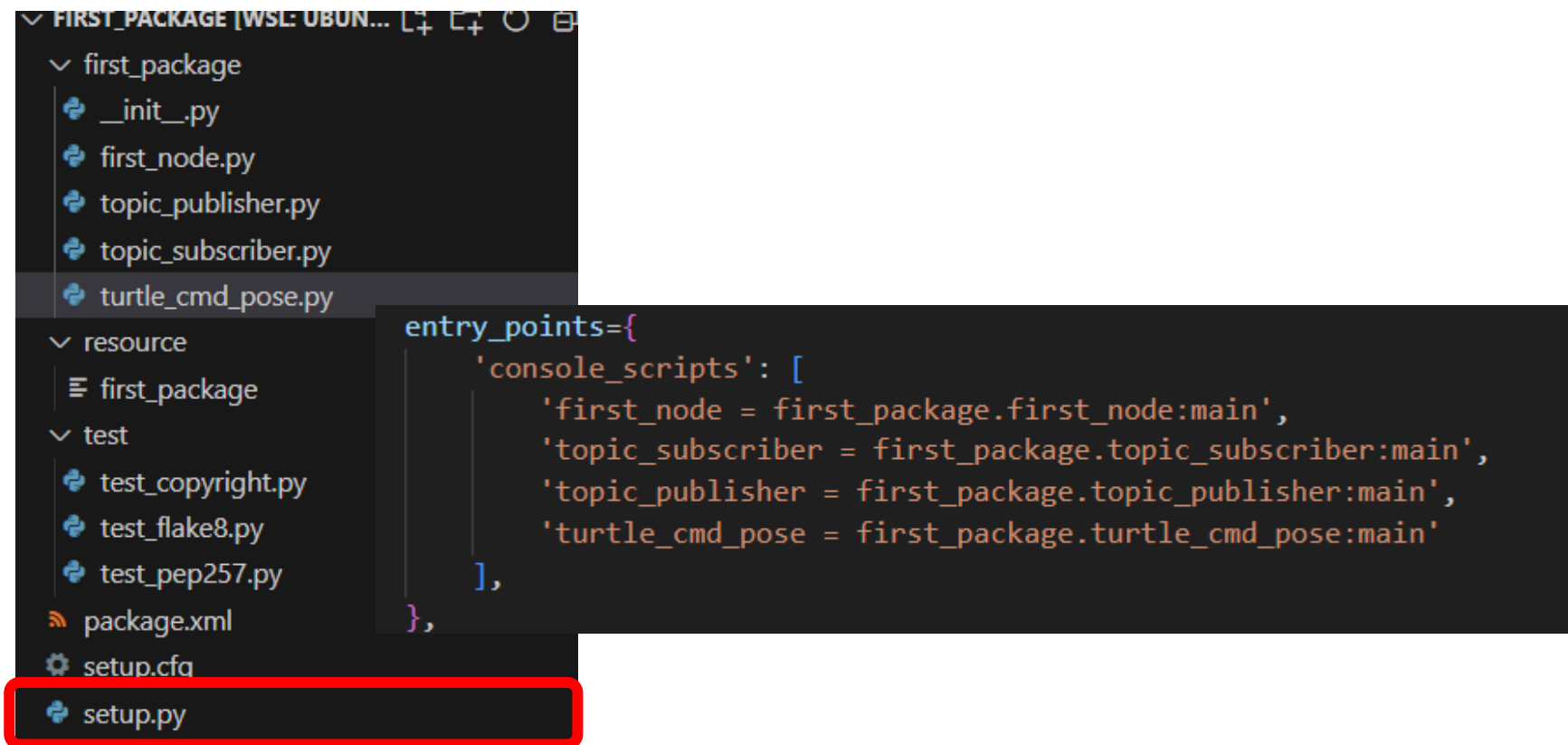
    turtle_cmd_pose.destroy_node()
    rp.shutdown()

if __name__ == '__main__':
    main()
```



❖ 정의된 Message 활용

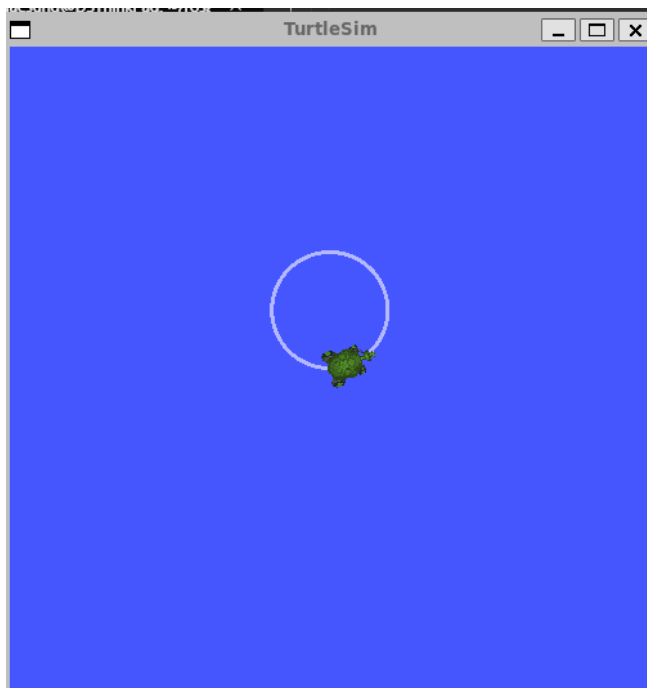
- 추가한 노드 실행을 위해서는 setup.py 수정 필요



```
entry_points={
    'console_scripts': [
        'first_node = first_package.first_node:main',
        'topic_subscriber = first_package.topic_subscriber:main',
        'topic_publisher = first_package.topic_publisher:main',
        'turtle_cmd_pose = first_package.turtle_cmd_pose:main'
    ],
}
```

❖ 정의된 Message 활용

- colcon build를 이용하여 다시 빌드한 후 turtlesim_node 실행 및 turtle_cmd_pose, topic_publisher 노드를 실행하여 동작 확인



```
daesung@DSThinkPad: ~/ros2_work
daesung@DSThinkPad: ~/ros2_work 38x14
QStandardPaths: wrong permissions on r
untime directory /run/user/1000/, 0755
instead of 0700
[INFO] [1727803673.347302677] [turtles
im]: Starting turtlesim with node name
/turtlesim
[INFO] [1727803673.354539214] [turtles
im]: Spawning turtle [turtle1] at x=[5
.544445], y=[5.544445], theta=[0.00000
0]
daesung@DSThinkPad: ~/ros2_work 80x14
308533, linear_velocity=2.0, angular_velocity=2.0)
turtlesim.msg.Pose(x=6.347755432128906, y=5.970935344696045, theta=0.94414699077
6062, linear_velocity=2.0, angular_velocity=2.0)
turtlesim.msg.Pose(x=6.365682601928711, y=5.997442245483398, theta=0.97614699602
12708, linear_velocity=2.0, angular_velocity=2.0)
turtlesim.msg.Pose(x=6.382751941680908, y=6.024509429931641, theta=1.00814700126
64795, linear_velocity=2.0, angular_velocity=2.0)
turtlesim.msg.Pose(x=6.398947238922119, y=6.0521087646484375, theta=1.0401469469
070435, linear_velocity=2.0, angular_velocity=2.0)
turtlesim.msg.Pose(x=6.41425085067749, y=6.080212116241455, theta=1.072147011756
897, linear_velocity=2.0, angular_velocity=2.0)
turtlesim.msg.Pose(x=6.428647518157959, y=6.108790397644043, theta=1.10414695739
7461, linear_velocity=2.0, angular_velocity=2.0)
```

❖ 정의된 Message 활용

- 앞서 turtlesim_node의 pose 구독을 확인했음
- 사용자가 설계한 message를 이용하여 토픽 메시지 처리
- turtle_cmd_pose.py파일에 다음과 같은 코드 추가

```
4 from first_package_msg.msg import FirstMsg
```

- callback_pose 메서드를 다음과 같이 수정

```
def callback_pose(self, msg):  
    self.cmd_pose.pose_x = msg.x  
    self.cmd_pose.pose_y = msg.y  
    self.cmd_pose.linear_vel = msg.linear_velocity  
    self.cmd_pose.angular_vel = msg.angular_velocity  
    print(self.cmd_pose)
```

❖ 정의된 Message 활용

- 수정이 완료된 패키지를 선택하여 빌드

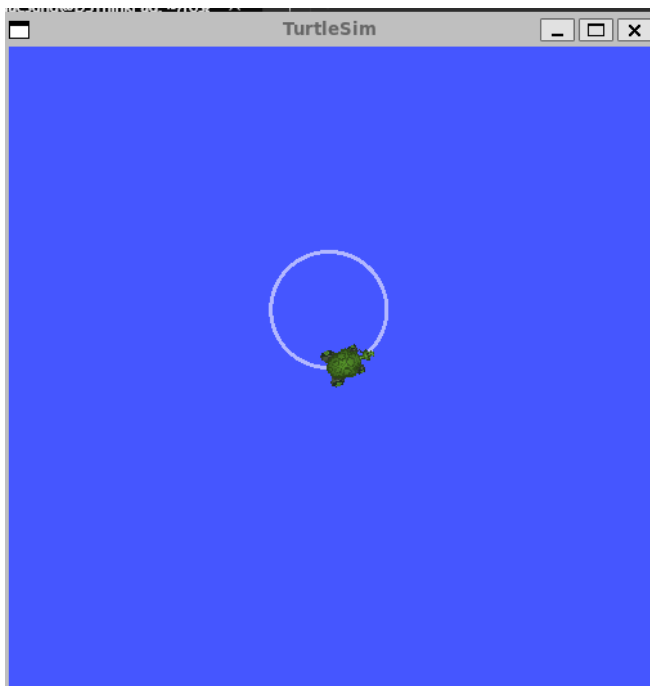
```
colcon build --packages-select first_package
```

```
daesung@DSThinkPad:~/ros2_work$ colcon build --packages-select first_package
Starting >>> first_package
Finished <<< first_package [0.65s]

Summary: 1 package finished [0.94s]
daesung@DSThinkPad:~/ros2_work$ |
```

❖ 정의된 Message 활용

- turtlesim_node 실행 및 turtle_cmd_pose , topic_publisher 노드를 실행하여 동작 확인



```

daesung@DSThinkPad: ~/ros2_work
daesung@DSThinkPad: ~/ros2_work 38x13
daesung@DSThinkPad:~/ros2_work$ ros2 r
un turtlesim turtlesim_node
QStandardPaths: wrong permissions on r
untime directory /run/user/1000/, 0755
instead of 0700
[INFO] [1727804927.363341877] [turtles
im]: Starting turtlesim with node name
/turtlesim
[INFO] [1727804927.369954893] [turtles
im]: Spawning turtle [turtle1] at x=[5
.544445], y=[5.544445], theta=[0.00000
0]
daesung@DSThinkPad: ~/ros2_work 80x13
first_package_msg.msg.FirstMsg(cmd_vel_linear=0.0, cmd_vel_angular=0.0, pose_x=5
.488821029663086, pose_y=5.545101642608643, linear_vel=2.0, angular_vel=2.0)
first_package_msg.msg.FirstMsg(cmd_vel_linear=0.0, cmd_vel_angular=0.0, pose_x=5
.520812034606934, pose_y=5.544345855712891, linear_vel=2.0, angular_vel=2.0)
first_package_msg.msg.FirstMsg(cmd_vel_linear=0.0, cmd_vel_angular=0.0, pose_x=5
.5528106689453125, pose_y=5.544613361358643, linear_vel=2.0, angular_vel=2.0)
first_package_msg.msg.FirstMsg(cmd_vel_linear=0.0, cmd_vel_angular=0.0, pose_x=5
.584784984588623, pose_y=5.545904636383057, linear_vel=2.0, angular_vel=2.0)
first_package_msg.msg.FirstMsg(cmd_vel_linear=0.0, cmd_vel_angular=0.0, pose_x=5
.616701126098633, pose_y=5.548218727111816, linear_vel=2.0, angular_vel=2.0)
first_package_msg.msg.FirstMsg(cmd_vel_linear=0.0, cmd_vel_angular=0.0, pose_x=5
.648527145385742, pose_y=5.5515522956848145, linear_vel=2.0, angular_vel=2.0)
  
```

❖ 정의된 Message 활용

- 앞서 turtlesim_node의 pose를 구독하여 사용자가 정의한 message에 저장됨을 확인
- cmd_vel 토픽도 구독하여 저장
- turtle_cmd_pose.py파일에 다음과 같은 코드 추가

```
5 from geometry_msgs.msg import Twist
```

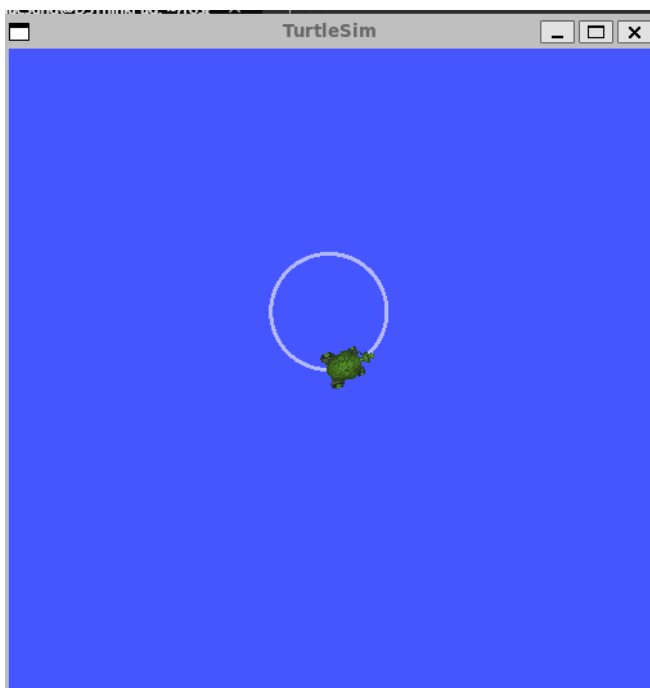
```
def __init__(self):  
    super().__init__('turtle_cmd_pose_node')  
    self.sub_pose = self.create_subscription(Pose, '/turtle1/pose', self.callback_pose, 10)  
    self.sub_cmd = self.create_subscription(Twist, 'turtle1/cmd_vel', self.callback_cmd, 10)  
    self.cmd_pose = FirstMsg()
```

- callback_cmd 메서드를 다음과 같이 작성

```
def callback_cmd(self, msg):  
    self.cmd_pose.cmd_vel_linear = msg.linear.x  
    self.cmd_pose.cmd_vel_angular = msg.angular.z  
    print(self.cmd_pose)
```

❖ 정의된 Message 활용

- 수정이 완료된 패키지를 선택하여 빌드
- turtlesim_node 실행 및 turtle_cmd_pose , topic_publisher 노드를 실행하여 동작 확인



```
daesung@DSThinkPad: ~/ros2_work
QStandardPaths: wrong permissions on runtime directory /run/user/1000/, 0755 instead of 0700
[INFO] [1727805600.716602434] [turtlesim]: Starting turtlesim with node name /turtlesim
[INFO] [1727805600.721938968] [turtlesim]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]

daesung@DSThinkPad: ~/ros2_work 80x11
first_package_msg.msg.FirstMsg(cmd_vel_linear=2.0, cmd_vel_angular=2.0, pose_x=4.592831611633301, pose_y=6.897505760192871, linear_vel=2.0, angular_vel=2.0)
first_package_msg.msg.FirstMsg(cmd_vel_linear=2.0, cmd_vel_angular=2.0, pose_x=4.721022129058838, pose_y=5.9543137550354, linear_vel=2.0, angular_vel=2.0)
first_package_msg.msg.FirstMsg(cmd_vel_linear=2.0, cmd_vel_angular=2.0, pose_x=5.612636566162109, pose_y=5.5478668212890625, linear_vel=2.0, angular_vel=2.0)
first_package_msg.msg.FirstMsg(cmd_vel_linear=2.0, cmd_vel_angular=2.0, pose_x=6.408684730529785, pose_y=6.069739818572998, linear_vel=2.0, angular_vel=2.0)
first_package_msg.msg.FirstMsg(cmd_vel_linear=2.0, cmd_vel_angular=2.0, pose_x=6.4072651863098145, pose_y=7.021602153778076, linear_vel=2.0, angular_vel=2.0)
```


❖ 정의된 Message 활용

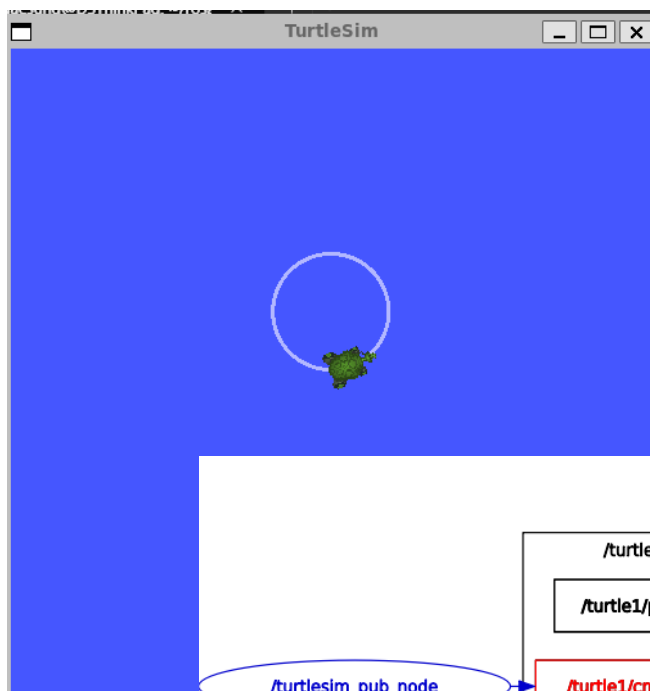
- 사용자가 정의한 message에 저장된 2개의 구독 결과를 발행
- turtle_cmd_pose.py파일에 다음과 같은 코드 추가

```
def __init__(self):
    super().__init__('turtle_cmd_pose_node')
    self.sub_pose = self.create_subscription(Pose, '/turtle1/pose', self.callback_pose, 10)
    self.sub_cmd = self.create_subscription(Twist, 'turtle1/cmd_vel', self.callback_cmd, 10)
    self.pub = self.create_publisher(FirstMsg, '/cmd_pose', 10)
    self.timer = self.create_timer(1.0, self.callback_timer)
    self.cmd_pose = FirstMsg()
```

```
def callback_timer(self):
    self.pub.publish(self.cmd_pose)
```

❖ 정의된 Message 활용

- 수정이 완료된 패키지를 선택하여 빌드
- turtlesim_node 실행 및 turtle_cmd_pose , topic_publisher 노드를 실행하여 동작 확인



```
daesung@DSThinkPad: ~/ros2_work
QStandardPaths: wrong permissions on runtime directory /run/user/1000/, 0755 instead of 0700
[INFO] [1727805600.716602434] [turtlesim]: Starting turtlesim with node name /turtlesim
[INFO] [1727805600.721938968] [turtlesim]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]

daesung@DSThinkPad: ~/ros2_work 38x11
daesung@DSThinkPad:~/ros2_work$ ros2 run first_package topic_publisher

daesung@DSThinkPad: ~/ros2_work 80x11
first_package msg.msg.FirstMsg(cmd_vel_linear=2.0, cmd_vel_angular=2.0, pose_x=4.592831611633301, pose_y=6.897505760192871, linear_vel=2.0, angular_vel=2.0)
md_vel_linear=2.0, cmd_vel_angular=2.0, pose_x=4.37550354, linear_vel=2.0, angular_vel=2.0)
md_vel_linear=2.0, cmd_vel_angular=2.0, pose_x=5.68212890625, linear_vel=2.0, angular_vel=2.0)
md_vel_linear=2.0, cmd_vel_angular=2.0, pose_x=6.9818572998, linear_vel=2.0, angular_vel=2.0)
md_vel_linear=2.0, cmd_vel_angular=2.0, pose_x=6.02153778076, linear_vel=2.0, angular_vel=2.0)
```

❖ 정의된 Message 활용

- 다음과 같이 동작하고 있음을 확인할 수 있음

rqt_topic__TopicPlugin - rqt

Topic Monitor

Topic	Type	Bandwidth	Hz	Value
<input checked="" type="checkbox"/> /cmd_pose	first_package_msg/msg/FirstMsg	unknown	1.00	
cmd_vel_linear	float			2.0
cmd_vel_angular	float			2.0
pose_x	float			6.467172145843506
pose_y	float			6.88914680480957
linear_vel	float			2.0
angular_vel	float			2.0
<input type="checkbox"/> /parameter_events	rcl_interfaces/msg/ParameterEvent			not monitored
<input type="checkbox"/> /rosout	rcl_interfaces/msg/Log			not monitored
<input checked="" type="checkbox"/> /turtle1/cmd_vel	geometry_msgs/msg/Twist	unknown	1.96	
linear	geometry_msgs/Vector3			
x	double			2.0
y	double			0.0
z	double			0.0
angular	geometry_msgs/Vector3			
x	double			0.0
y	double			0.0
z	double			2.0
<input checked="" type="checkbox"/> /turtle1/pose	turtlesim/msg/Pose	unknown	62.36	
x	float			5.627089023590088
y	float			7.539525032043457
theta	float			3.0267930030822754
linear_velocity	float			2.0
angular_velocity	float			2.0
<input type="checkbox"/> /turtle1/color_sensor	turtlesim/msg/Color			not monitored
<input type="checkbox"/> /turtle1/rotate_absolute/_action/feedback	turtlesim/action/RotateAbsolute_FeedbackMessage			can not get message cla...
<input type="checkbox"/> /turtle1/rotate_absolute/_action/status	action_msgs/msg/GoalStatusArray			not monitored

❖ Mission

1. 사용자가 주고 받을 메시지 만들기
 1. 랜덤주행 시작, 정지
 2. 랜덤주행 영역지정 좌표(4개)
2. 2개의 노드(node1, node2)를 생성하고 node1은 turtlesim_node로 랜덤주행 Topic을 발행하며 node2로 부터 사용자가 정의한 메시지를 수신하여 랜덤주행 정보에 적용
3. Node2는 사용자가 값을 설정하고 사용자가 정의한 메시지를 이용하여 node1으로 Topic 발행

Q & A
