

ROS2\_day1 hw3 ROS2 python/C++  
통신 과제 보고서

로봇 20기 인턴 현장석

## 목차

- 코드 설명

## - 코드 설명

```
1  import rclpy
2  from rclpy.node import Node
3  from std_msgs.msg import String, Float64
4  from turtlesim.srv import SetPen
5  from geometry_msgs.msg import Twist
6  import sys
7  import termios
8  import tty
9  import time
10
11  def get_key():
12      fd = sys.stdin.fileno()
13      old_settings = termios.tcgetattr(fd)
14      try:
15          tty.setraw(sys.stdin.fileno())
16          ch = sys.stdin.read(1)
17      finally:
18          termios.tcsetattr(fd, termios.TCSADRAIN, old_settings)
19      return ch
20
21  def Turtle_color_width(node, a, b, c, d):
22      client = node.create_client(SetPen, '/turtle1/set_pen')
23      req = SetPen.Request()
24      req.r = a
25      req.g = b
26      req.b = c
27      req.width = d
28      req.off = 0
29      client.call_async(req)
30
31  class TurtlePublisher(Node):
32      def __init__(self):
33          super().__init__('cmd_vel')
34          self.TurtlePublisher = self.create_publisher(Twist, 'turtle1/cmd_vel', 10)
35          # self.Turtle_color_width_Publisher = self.create_publisher(Twist, 'turtle1/cmd_vel', 10)
36
37
38
39  def publish_move(self):
40      while(1) :
41          key = get_key()
42          twist = Twist()
```

```
40 while(1) :
41     key = get_key()
42     twist = Twist()
43
44     if key == 'w':
45         twist.linear.x = 2.0
46     elif key == 's':
47         twist.linear.x = -2.0
48     elif key == 'a':
49         twist.angular.z = 2.0
50     elif key == 'd':
51         twist.angular.z = -2.0
52     elif key == '1' :
53         Turtle_color_width(self,255, 0, 0,7)
54
55     for i in range(4) :
56         if(i==0) :
57             twist.linear.x = 2.0
58
59         if(i==1) :
60             twist.linear.x = 0.0
61             twist.linear.y = 2.0
62
63         if(i==2) :
64             twist.linear.x = -2.0
65             twist.linear.y = 0.0
66         if(i==3) :
67             twist.linear.y = -2.0
68             twist.linear.x = 0.0
69         self.TurtlePublisher.publish(twist)
70         time.sleep(0.5)
71     continue
72
73
74
75     elif key == '2' :
76         Turtle_color_width(self,0, 0, 255,10)
```

---

```

39         def publish_move(self):
73
74
75         elif key == '2' :
76             Turtle_color_width(self,0, 0, 255,10)
77             for i in range(5) :
78                 if(i==0) :
79
80                     twist.linear.x = 2.0
81
82                 if(i==1) :
83                     twist.linear.x = 2.0
84                     twist.linear.y = 2.0
85
86                 if(i==2) :
87                     twist.linear.x = -2.0
88                     twist.linear.y = 0.0
89
90                 if(i==3) :
91                     twist.linear.x = -2.0
92
93                 if(i==4) :
94                     twist.linear.x = 2.0
95                     twist.linear.y = -2.0
96
97                 self.TurtlePublisher.publish(twist)
98                 time.sleep(0.5)
99                 continue
100
101
102
103         elif key == '3' :
104             Turtle_color_width(self,0, 255, 0,17)
105
106
107             twist.linear.x = 2.0
108             twist.angular.z = 2.0
109
110
111
112             self.TurtlePublisher.publish(twist)
113
114
115     def main(args=None):
116         rclpy.init(args=args)
117         node = TurtlePublisher()
118         try :
119             node.publish_move()
120
121         except KeyboardInterrupt:
122             node.get_logger().info('Keyboard interrupt (SIGINT)')
123         finally:
124             node.destroy_node()
125             rclpy.shutdown()
126     if __name__ == '__main__':
127         main()

```

클라이언트 라이브러리, 노드 생성, 초기화, 섯다운 등에 쓰이는 ros2 모듈을 불러온다. 사용자가 만든 클래스가 Node를 상속받아 퍼블리셔/ 서브스크라이버 기능을 사용하기 위해 ros2의 노드 클래스를 불러왔다.

도형을 그릴 때 펜의 색, 두께 등을 변경하기 위해 turtlesim 패키지의 서비스 기능 중 setpen을 불러왔다. 터틀을 이동시키기 위한 Twist 를 불러왔다. sys를 임포트한 이유는 키보드 입력을 받기 위해서 이다.

termios는 터미널 의 속성을 제어하는 모듈로 특정 키 입력으로 터틀을 움직이게 하기 위해 사용해야였다. import tty 로 키 입력을 즉시 받을 수 있게 하였다. 터틀이 도형을 그릴 때 출력 명령의 간격이 너무 짧아 선만 계속 출력되는 버그가 발생하여서 time을 임포트하여 시간 지연 sleep을 사용하였다. def key 는 거북이를 움직이도록 명령을 내리기 위해 키보드 입력 값을 받는 함수이다 . 인터넷에서 참고하였다.

한 글자(1바이트)를 읽고 ch에 저장한다. 출력이 비워질 때까지 기다린 뒤 읽었던 한 글자 키 값을 리턴한다.

def Turtle\_color\_width 함수는 터틀의 펜 색상과 두께를 설정하기 위한 함수이다. setpen 서비스를 호출하기 위해서 클라이언트를 선언하였다. 서비스의 이름은 /turtle1/set\_pen이다. SetPen.Request()

SetPen 서비스 요청 메시지를 생성한 후 rgb값으로 펜의 색깔과 width로 두께를 설정할 수 있게 하였다. 노드 이름을 cmd\_vel 이라고 정의한 후 turtle1/cmd\_vel' 토픽에 발행하는 퍼블리셔를 생성하였다.

wasd에 따라 터틀이 움직이도록 publish\_move 메서드에서 정의하였고 숫 자123 키를 입력했을 때 사각형, 삼각형, 원이 그려지도록 하였다. time.sleep(0.5) 위에서 언급한 버그가 발생하지 않도록 해주었다 . rclpy.init(args=args)ROS 2 파이썬 클라이언트 초기화 해주고 node = TurtlePublisher() 위에서 정의한 TurtlePublisher 노드 인스턴스를 생성한다.

