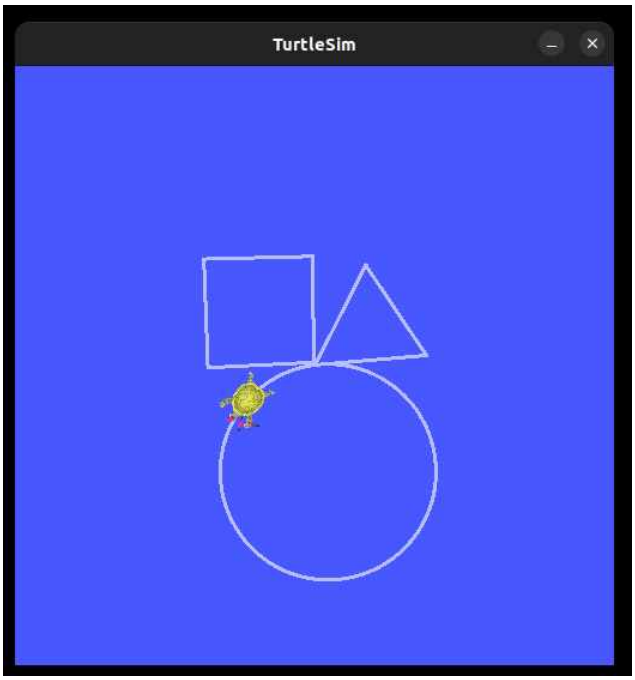
hw1 보고서

20기 인턴 송수민

1) 실행 결과



2) 토픽 통신 보고서

1. 토픽(topic)의 개념

토픽은 ROS에서 메시지 전달 통로 역할을 한다.

노드(node)는 토픽을 통해 서로 데이터를 주고받는다.

- 두 가지 역할

Publisher(발행자): 토픽에 메시지를 발행

Subscriber(구독자): 토픽을 구독하고 메시지를 수신

2. 통신 방식

ROS의 토픽 통신은 비동기(pub-sub) 방식

발행자는 메시지를 특정 토픽에 계속 보낸다.

구독자는 해당 토픽을 듣고 있다가 새로운 메시지가 도착하면 받아 처리한다.

발행자와 구독자는 서로 직접 연결되지 않고, ROS 마스터(roscore)를 통해 관리된다.

-> 즉, 발행자와 구독자는 서로를 몰라도, 같은 토픽 이름과 메시지 타입만 맞으면 통신이 가능하다.

3. turtlesim에서의 예시
노드 실행
turtlesim_node : 거북이를 화면에 띄우는 노드
ros2 topic pub --once : 사용자가 한 번만 토픽에 메시지를 발행하는 명령어

주요 토픽
/turtle1/cmd_vel → 발행자: ros2 topic pub --once 명령어, 구독자: turtlesim_node

메시지 구조
ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist ₩
"{linear: {x: 2.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 1.8}}"

-> 직선 속도(linear.x = 2.0)와 회전 속도(angular.z = 1.8)를 동시에 정의하여, 거북이가 앞으로 이동하면서 회전하도록 함

/turtle1/pose : turtlesim/msg/Pose
x: 5.54
y: 5.54
theta: 0.0
linear_velocity: 0.0
angular_velocity: 0.0

-> 현재 거북이의 위치와 자세

4. 동작 흐름 요약
1) 사용자가 ros2 topic pub 명령어로 /turtle1/cmd_vel 토픽에 Twist 메시지 발행
2) turtlesim_node가 해당 토픽을 구독하여 거북이 움직임 반영
3) 동시에 turtlesim_node는 /turtle1/pose 토픽에 위치 메시지 발행
4) 다른 노드(또는 ros2 topic echo)가 이를 구독하여 거북이의 좌표를 읽을 수 있음

- 토픽 통신 구조 = 발행자 ↔ 토픽 ↔ 구독자

5. 정리
ROS2에서 토픽은 노드 간 데이터 교환 통로
발행자와 구독자는 서로 독립적이며, ROS 2의 미들웨어(DDS)가 연결을 중재
turtlesim 실습을 통해, 우리가 그린 원·사각형·삼각형 궤적도 결국 속도 메시지를 발행 -> 궤적 형성이라는 토픽 통신 원리 덕분에 가능했음