

1. 학습 목적

두 번째 과제는 day1에서 생성한 turtlesim 조종 패키지를 qt의 gui와 연결하여 ui 명령을 통해서 ros2의 turtlesim_node 와 통신하는 것을 목적으로 한다. 그리고 조종 명령을 담고 있는 cmd_vel 값을 gui에 출력하고자 한다.

2. 과제 구현

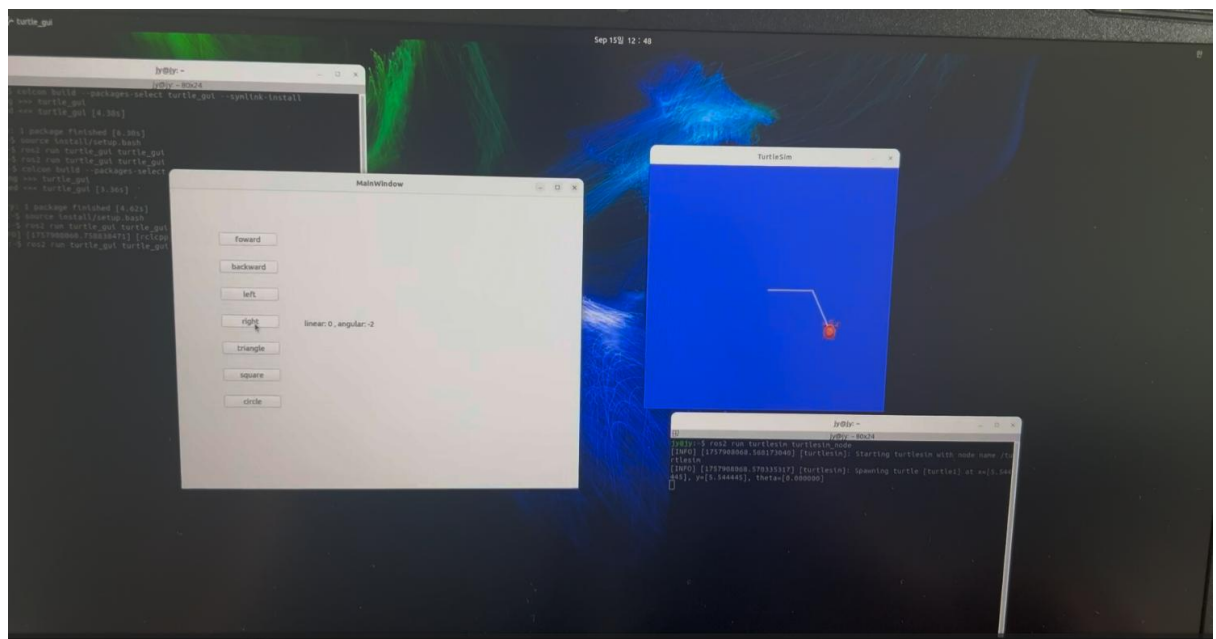
```
jy@jy:~/ros2_ws/src/turtle_gui$ tree
.
├── CMakeLists.txt
├── include
│   └── turtle_gui
│       └── mainwindow.hpp
├── package.xml
├── resource
│   └── mainwindow.ui
└── src
    ├── main.cpp
    └── mainwindow.cpp

4 directories, 6 files
```

우선 패키지의 구조를 다음과 같이 생성하였다. Ui 구성의 경우 직진, 후진, 좌회전, 우회전, 삼각형 그리기, 사각형 그리기, 원 그리기 기능을 구현할 총 7개의 버튼과 cmd_vel 값을 표시한 label 하나로 구성하였다.

GUI 내에서 버튼을 누르면 함수를 호출하여 좌표로 (linear, angular)값을 나타낸 메시지를 생성하고 이를 publisher로 송신하도록 하였다. 그리고 도형의 경우 day1에서 사용하였던 알고리즘을 그대로 차용하여 사각형은 직진 후 회전을 4번 반복, 삼각형의 경우 3번, 원의 경우 일정 시간동안 지속하여 회전하도록 하여 구현하였다. CMakeLists.txt는 계속해서 빌드 오류가 나서 어쩔 수 없이 수동으로 경로 지정 (qt5_wrap_ui, qt5_wrap_cpp) 방식으로 설정하였다.

3. 실행 결과



다음은 각각의 터미널에서 turtlesim_node와 turtle_gui 노드를 실행하여 과제를 구현하고 있는 사진이다. 버튼에 따라 동작하고 cmd_vel 값이 표시되는 것을 볼 수 있다.