

# ROS2\_2 일차\_HW\_02\_01\_REPORT

19 기 예비단원 최범수

이 과제는 1 일차의 거북이 관련 조종, 색, 두께 등을 qt 로 UI 를 구성해 버튼을 누르는 방식으로 turtlesim\_node 를 조절하는 것이다. 관련 설정을 위해 우선 패키지를 작성하였고,

```
#include <rclcpp/rclcpp.hpp>
#include <QObject>
#include <QThread>
#include <geometry_msgs/msg/twist.hpp>
#include <std_srvs/srv/empty.hpp>
#include <rclcpp/parameter_client.hpp> // 파라미터 관련 서비스
#include <turtlesim/srv/set_pen.hpp>
```

프로그램 작성을 위해 여러 설정들을 include 해 UI 상에서 조절이 가능하게 했다. 코드 관련 설명은 주석으로 대체하였으니 각 함수명들의 기능을 소개하겠다.

```
void MainWindow::changeLineWidth(int width) {
qnode->PenWidth(width); // QNode 의 펜 두께 함수 호출
}
```

펜의 두께를 조절하는 함수이다.

```
void MainWindow::closeEvent(QCloseEvent* event) {
QMainWindow::closeEvent(event);
}
```

속도를 조절해 wasd 버튼을 통해 조절이 가능하다

```
void MainWindow::on_W_clicked() {
qnode->publishVelocity(1.0, 0.0); // 앞으로 이동
}
```

```
void MainWindow::on_S_clicked() {
qnode->publishVelocity(-1.0, 0.0); // 뒤로 이동
}
```

```
void MainWindow::on_A_clicked() {
qnode->publishVelocity(0.0, 1.0); // 왼쪽 회전
}
```

```
void MainWindow::on_D_clicked() {
qnode->publishVelocity(0.0, -1.0); // 오른쪽 회전
}
```

// 배경색 변경 함수들

```
void MainWindow::BackgroundColorRed() {
qnode->backgroundColor(255, 0, 0); // 빨강
}
```

```
void MainWindow::BackgroundColorGreen() {
qnode->backgroundColor(0, 255, 0); // 초록
}
```

```
}
```

```
void MainWindow::BackgroundColorBlue() {  
qnode->backgroundColor(0, 0, 255); // 파랑  
}
```

선의 색을 조절 가능하다.

```
void MainWindow::lineColorRed() {  
qnode->lineColor(255, 0, 0); // 빨강  
}
```

```
void MainWindow::lineColorGreen() {  
qnode->lineColor(0, 255, 0); // 초록  
}
```

```
void MainWindow::lineColorBlue() {  
qnode->lineColor(0, 0, 255); // 파랑  
}
```

cmd\_vel 의 값을 GUI 에 출력하도록 설정하였다.

```
void MainWindow::displayCmdVel(QString data) {  
ui->text->append(data); // "text"에 데이터 추가  
}
```

우선 색상과 펜의 두께는 서로 상호작용 가능해서 색상을 변경해도 기본 두께를 current\_r,g,b 등의 변수를 통해 전달받는다. 처음 두께를 변경했을 때 기본 색이 지정되어야 해서 빨간색으로 지정을 해두었고 이로 인해 처음 두께를 조절하면 빨간색으로 변한다. 색을 변경한 뒤 두께를 설정하면 문제가 없다.

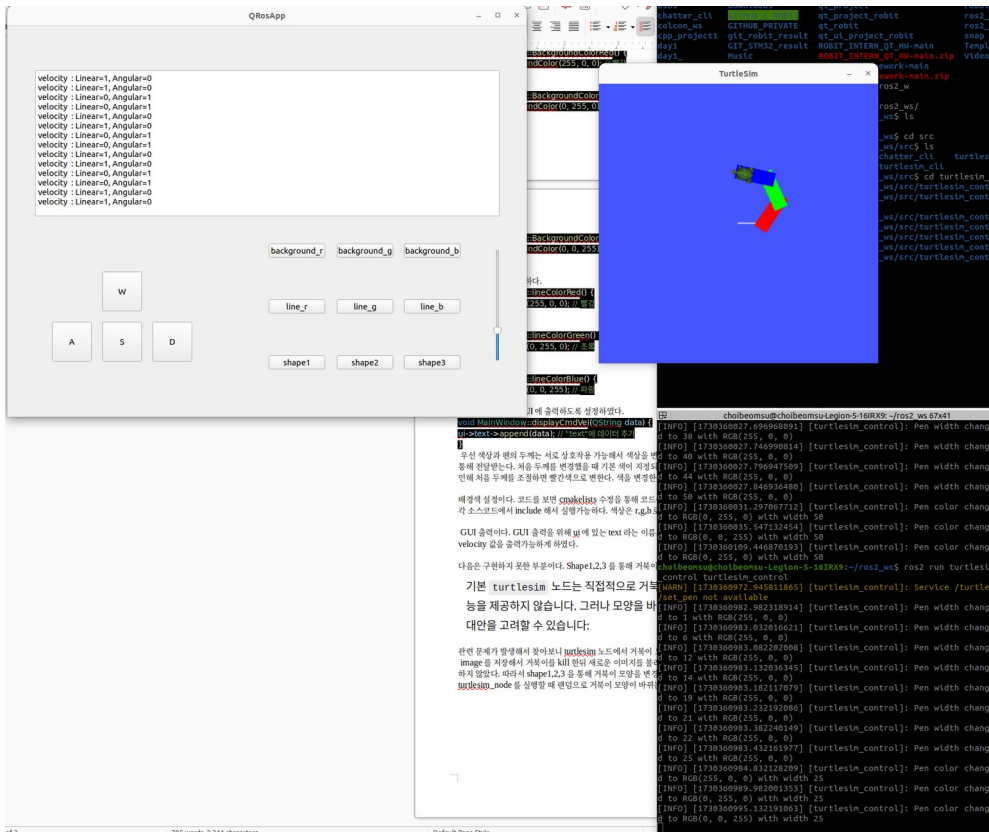
배경색 설정이다. 코드를 보면 cmakeLists 수정을 통해 코드상에서 파라미터 조절이 가능하도록 추가했고, 각 소스코드에서 include 해서 실행가능하다. 색상은 r,g,b 로 제한해 두었다.

GUI 출력이다. GUI 출력을 위해 ui 에 있는 text 라는 이름의 QTextBrowser 를 이용해 현재 cmd\_vel 즉 velocity 값을 출력가능하게 하였다.

다음은 구현하지 못한 부분이다. Shape1,2,3 를 통해 거북이의 모양을 변경하고자 하였는데

기본 turtlesim 노드는 직접적으로 거북이 모양을 바꾸는 기능을 제공하지 않습니다. 그러나 모양을 바꾸고 싶다면 몇 가지 대안을 고려할 수 있습니다:

관련 문제가 발생해서 찾아보니 turtlesim 노드에서 거북이 모양을 바꾸는 기능을 제공하지 않는다고 나왔다. image 를 따로 저장해서 거북이를 kill 한뒤 인터넷에서 다운받은 새로운 이미지를 불러오는 방법이 있다. 그리고 circle, Triangle, Square 버튼을 통해 길이 조절이 가능하며 도형을 그리는 기능을 실행하였다. 각각의 함수명은 drawCircle, drawTriangle, drawSquare 이다.



실행화면 예시를 첨부하였다. GUI 상에 velocity 가 출력되고 wasd 조종이 가능하며 색 변경, 두께 설정이 가능하다.

배경색상 변경 또한 가능하다.

