ROS2 1일차 과제

19기 예비단원 이원준

# 과제 1 - 소스코드 로직 설명

* 키 입력 처리는 termios 헤더파일을 활용해서 현재 터미널 창의 상태를 oldt에 저장하고 버퍼링과 에코를 비활성화함으로써 터미널에서 글자를 버퍼링없이 즉시 입력받도록 구성했다.
* 콜백함수에서 키 처리를 하고 효율적인 구성을 위해 항상 토픽을 발행하게 한다.
* 펜 설정 함수는 펜 값을 get\_ 함수로 받아오고 비동기적으로 request 클라이언트 요청을 보낸다.
* 이후 콜백으로 설정된 익명함수로 respond를 받아오고 변경 여부를 사용자에게 알린다.
* 배경색 변경함수는 get\_color\_value 함수로 색상값을 받아오고 파라메터 클라이언트가 서비스가 준비되었는지 확인한다.
* 서비스 준비가 되면 파라메터를 설정해주는데 이때 이작업이 비동기 작업이므로 설정이 완료되면 익명함수를 콜백한다.
* 콜백함수에서 result의 각 항목을 확인하여 모든 파라메터 설정이 성공했는지 검사한다.
* 이때 설정이 하나라도 실패하면 success 변수에 거짓값을 넣는다.
* 이후 파라메터 설정이 성공하면 clear\_client\_를 통해 clear 서비스를 호출한다.
* 서비스가 호출됨에 따라 콜백함수를 호출하는데 respond를 통해 배경 변경여부를 사용자에게 알린다.
* 거북이 모양 선택 메소드 같은 경우 완성하지는 못했다.
* 기본적인 로직은 선택값을 받고 거북이를 삭제시킨 후 parameter\_client를 초기화한 상태에서 파라메터로 모양을 전송하고 다시 거북이 생성 요청을 서비스로 보내는것이 계획이였지만 디버깅이나 공부가 충분치 못한것 같다.

사진 1 . 펜 설정 메소드 및 키 처리 예시

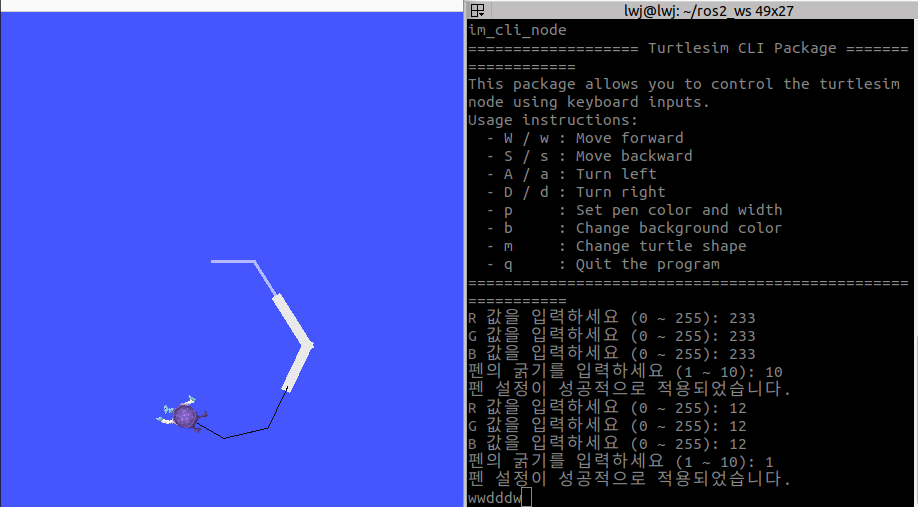


사진 2. 배경색 변경 메소드 예시 사진

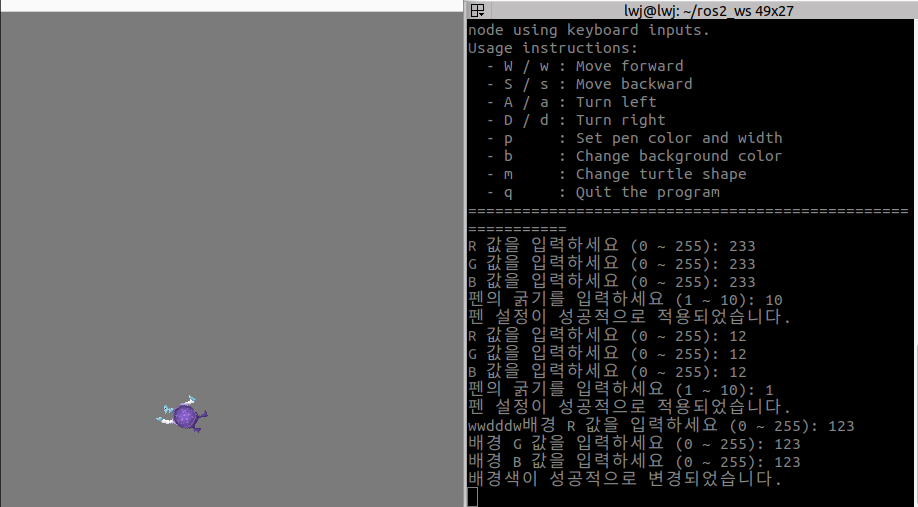
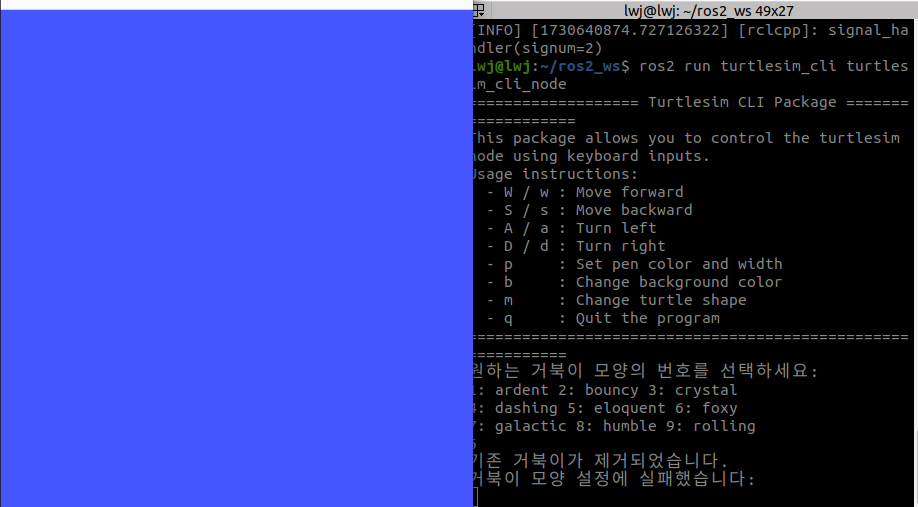


사진 3. 거북이 모양 변경 실패 사진



# 과제 2 - 소스코드 로직 설명

* 클라스 생성자에서 그릴 이미지를 사용자가 지정하고 이를 바탕으로 적절한 함수를 실행시키는 코드이다.
* 거북이 회전 각도는 호도법을 적용해 계산하였다.
* 원은 작은 움직임을 여러번하는 방법으로 자연스럽게 구성하였다.
* 거북이가 그림을 그릴때 벽과 부딪치면 그림이 비정상적으로 그려지는 한계가 있다.
* 처음 생성 좌표와 벽까지의 거리를 계산해서 그리는 모양마다 적절한 한계크기를 지정해 예외처리를 하고 싶었으나 그 부분은 실패했다.
* Cli의 예외처리 아쉽게 느껴진다.

사진 1. 벽과 부딪치는 거북이

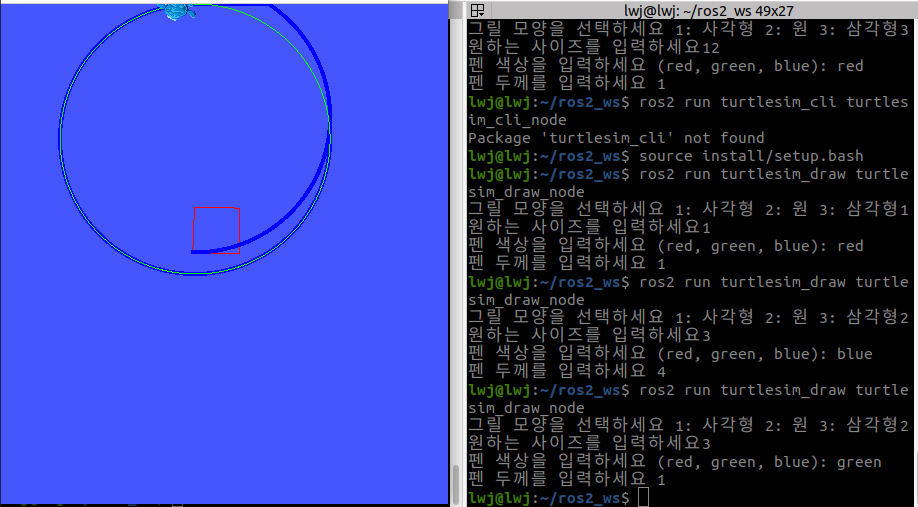
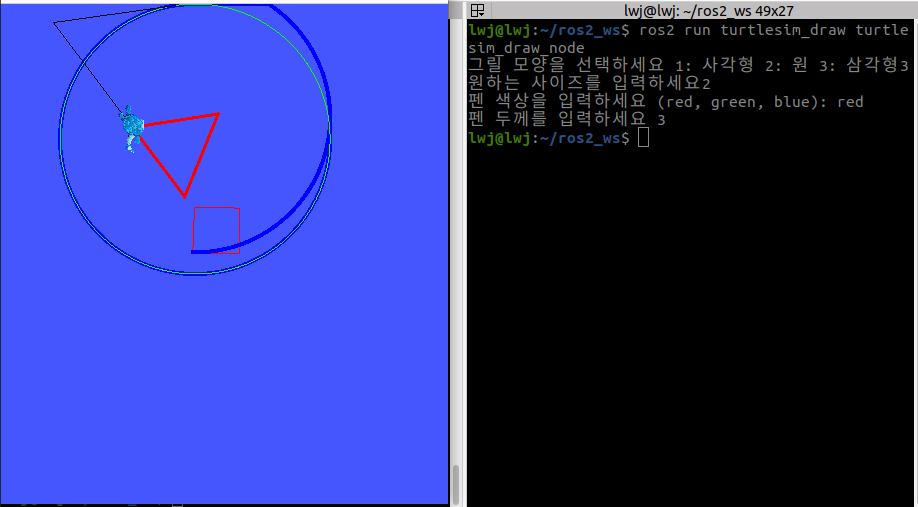


사진 2.



# 과제 3.

* Std 메세지의 문자열과 정수형(int64)을 활용해 두 노드 사이 토픽 통신을 한다.
* 콜백함수에서 토픽이 발행되는 횟수를 센다.
* 터미널 출력은 RCLCPP\_INFO을 활용한다.

사진 1.

