

가상 환경에서의 서빙 로봇 시스템 구현

실습 과제: turtlebot 3(가상환경 rviz2 및 gazebo)를 이용하여 서빙 로봇 시스템을 구현하여라. 손님 GUI와 주방 GUI를 만들어 로봇과 통신되는 시스템을 구축

제한 기간: 5일

필수 구현 요소 : ex) 일일 매출, 매뉴별 매출, 선호 매뉴

1. 테이블 오더를 통한 주문
2. 주방 디스플레이를 통한 주문 접수
3. 주방 디스플레이를 통한 서빙 로봇 제어
4. 서빙 로봇 작동
5. 주문 내역을 데이터 베이스에 저장
6. 데이터 베이스의 데이터를 이용한 통계
7. 메시지 인터페이스
8. Logging
9. QoS (Qualit of Service) 1개 이상 작성하여 프로젝트에 이용
10. 토픽, 서비스, 액션 각각 한 개 이상 사용

1. 사전 설치 요소들

```
sudo apt install ros-humble-gazebo-*
```

```
sudo apt install ros-humble-cartographer
```

```
sudo apt install ros-humble-cartographer-ros
```

```
sudo apt install ros-humble-navigation2
```

```
sudo apt install ros-humble-nav2-bringup
```

```
sudo apt install ros-humble-dynamixel-sdk
```

```
sudo apt install ros-humble-turtlebot3-msgs
```

```
sudo apt install ros-humble-turtlebot3
```

사전작업

```
mkdir -p ~/turtlebot3_ws/src/
```

```
cd ~/turtlebot3_ws/src/
```

```
git clone -b humble-devel https://github.com/ROBOTIS-GIT/turtlebot3.git
```

```
git clone -b humble-devel https://github.com/ROBOTIS-GIT/turtlebot3\_msgs.git
```

```
git clone -b humble-devel https://github.com/ROBOTIS-GIT/turtlebot3\_simulations.git
```

```
cd ~/turtlebot3_ws && colcon build --symlink-install
```

2. 환경 설정

```
echo 'source ~/turtlebot3_ws/install/local_setup.bash' >> ~/.bashrc
```

```
echo 'source /usr/share/gazebo/setup.sh' >> ~/.bashrc
```

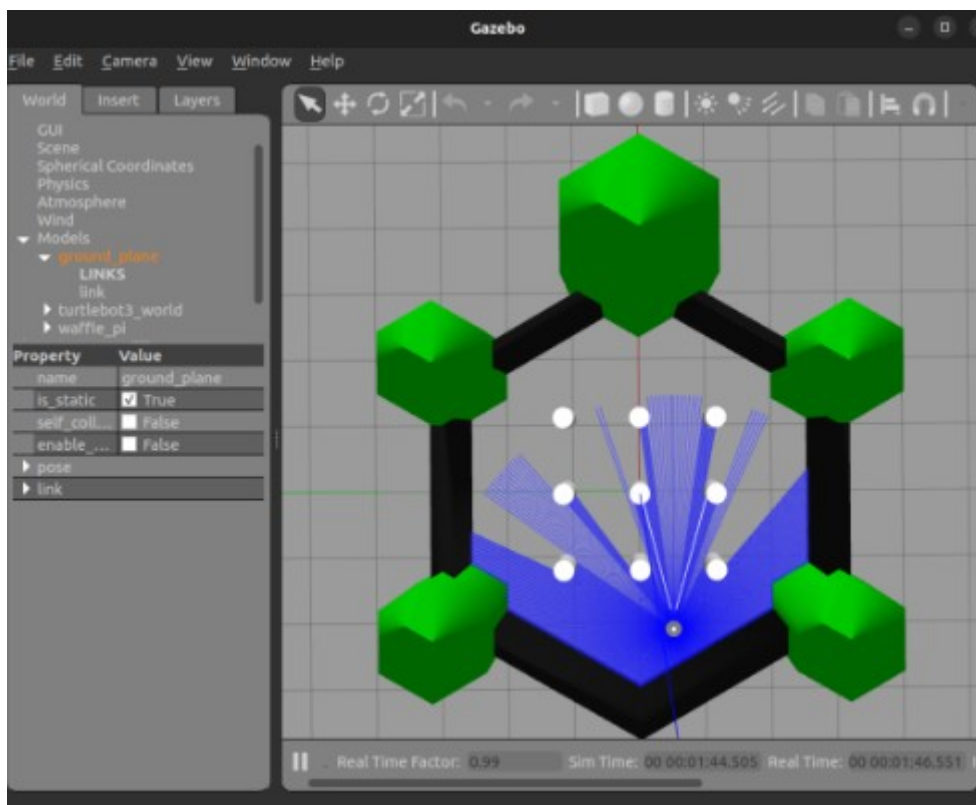
```
echo 'export TURTLEBOT3_MODEL=waffle_pi' >> ~/.bashrc
```

```
echo 'export ROS_LOCALHOST_ONLY=1 ' >> ~/.bashrc
```

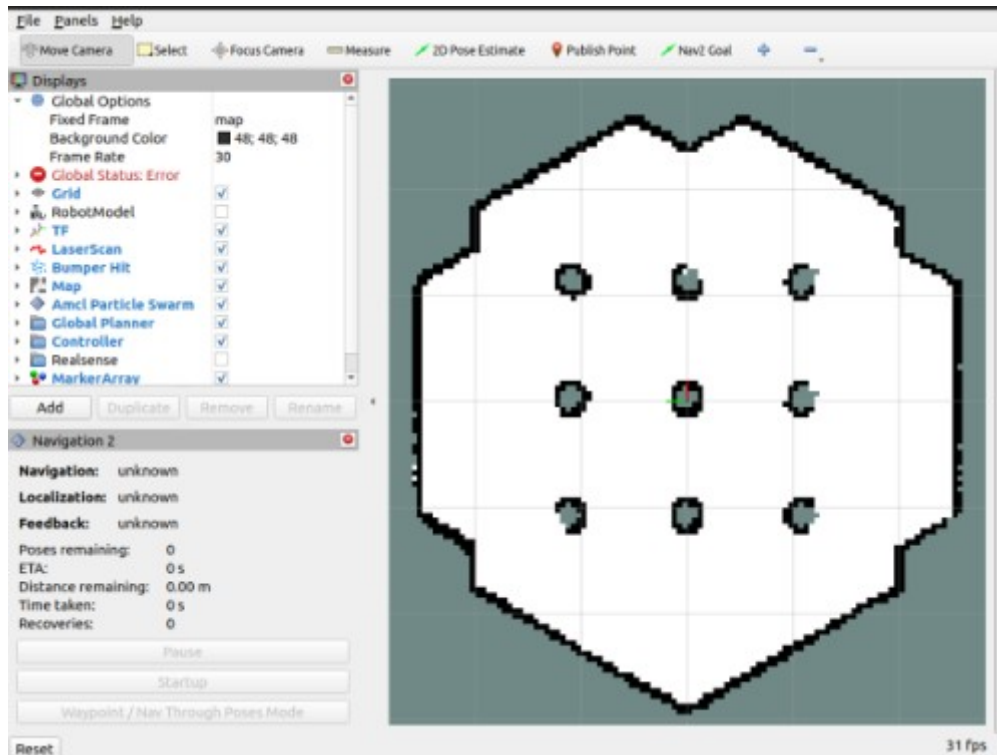
```
source ~/.bashrc
```

3. 가제보 실행

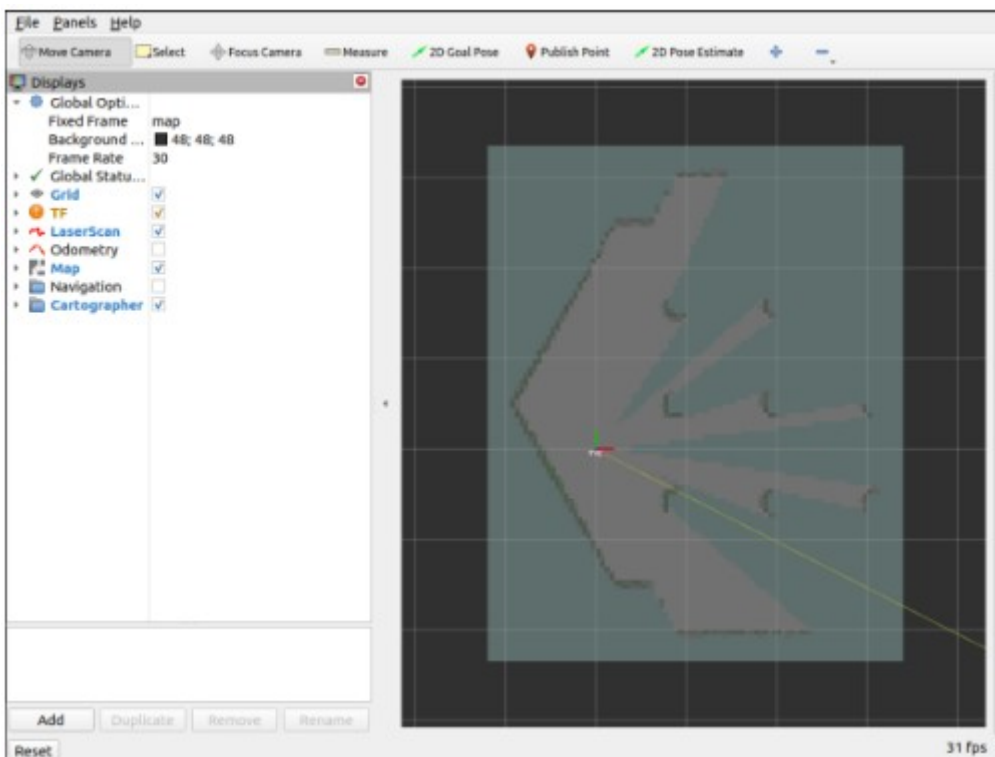
```
ros2 launch turtlebot3_gazebo turtlebot3_world.launch.py
```



4. nav2 실행(ros2 launch turtlebot3_navigation2 navigation2.launch.py)



5. 카토그래퍼 실행(ros2 launch turtlebot3_cartographer cartographer.launch.py)



6. 맵 저장(ros2 run nav2_map_server map_saver_cli -f~/map)



7. QT Designer

Qt Designer을 이용해 gui 디자인을 진행한다.

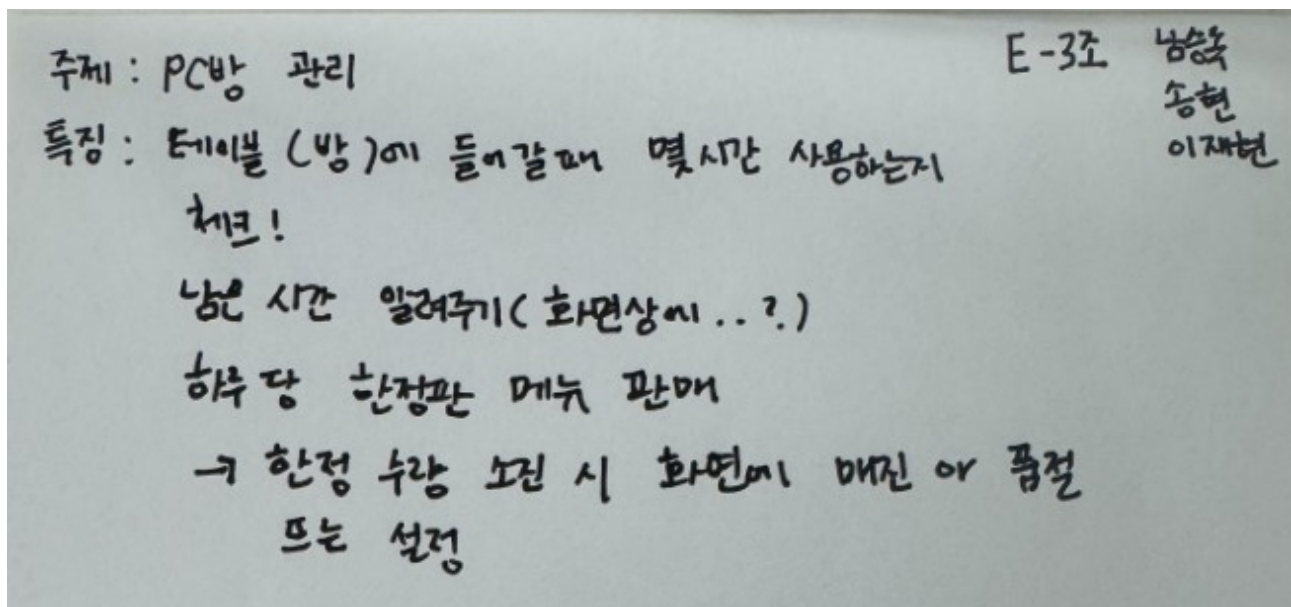
Qt Designer 설치

```
sudo apt update
```

```
sudo apt install qttools5-dev-tools
```

Qt Designer 실행

```
designer
```



<다른조와의 차별점을 생각하기 위한 우리팀의 초기회의>

→후에는 로봇상태 확인까지! 추가됨

export TURTLEBOT3_MODEL=burger(우리모델은 직각팔이 구우우우지 필요없기에 팔이 있는 waffle이 아니라 burger를 통해 진행했다)

위와 같은 알고리즘을 짜기 위해 PC_ORDER.py를 고객 부분으로 만들고
kitchen.py를 주방의 부분으로 만들었다.

여기서 주방은 로봇의 초기위치와 9개의 테이블의 위치의 정보가 필요하기에 각각의 rviz상 좌표가
필요했는데 그 정보는 ros2 topic echo/clicked_point로 알수 있었다.

```
header:
  stamp:
    sec: 1739751422
    nanosec: 179189071
  frame_id: map
point:
  x: 1.6802361011505127
  y: 0.7761558294296265
  z: 0.002471923828125
--
```

Publish point는 옆과 같이 나옴

<시연 유튜브 링크>

<https://www.youtube.com/watch?v=pXBTrAc8Bbg>