

생존자 구조 프로그램

[E-2] 배민지 서연우 진민혁



시나리오

원자력 발전소가 폭발한 위급한 상황에서,
터틀봇 3대가 방사선 위험 지역에 투입되어 생존자를 구조!

드론은 상공에서 현장 상황을 실시간으로 전달한다.

터틀봇과 드론의 협력 덕분에 구조팀은 생존자들을 무사히 구출한다

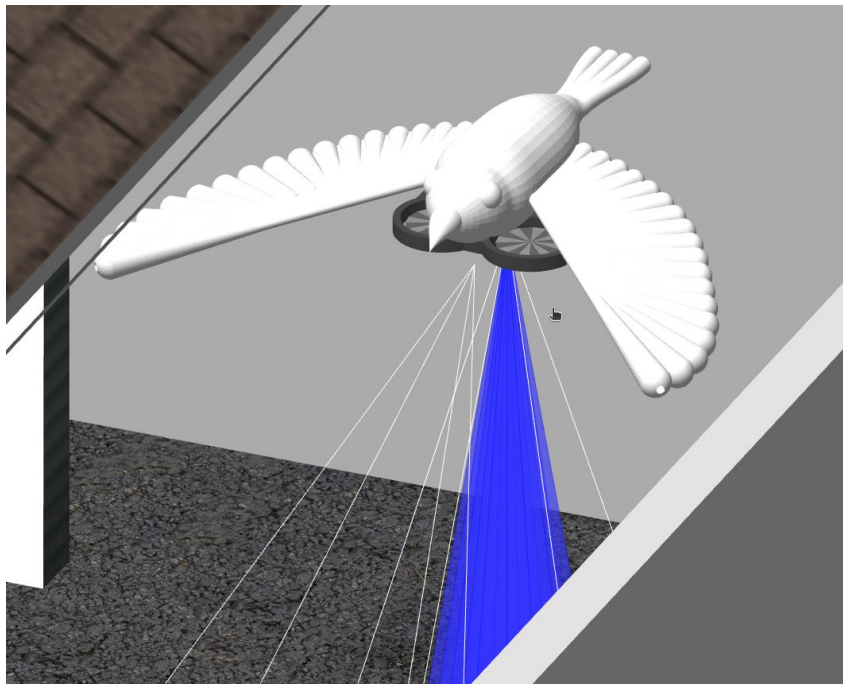
전체 구조

1. `multitb_interfaces`: 서비스 인터페이스 패키지
2. `py_srvcli`: 관제시스템 `flask` 생성 패키지
3. `turtlebot3_multi_robot`: 다중 로봇 launch, 모델 로드 패키지
4. `yolo_human_navi`: 생존자 yolo 감지, 터틀봇 내비게이션
5. `sjtu_drone`: 드론 패키지 (github)

사용한 협동 로봇 모델

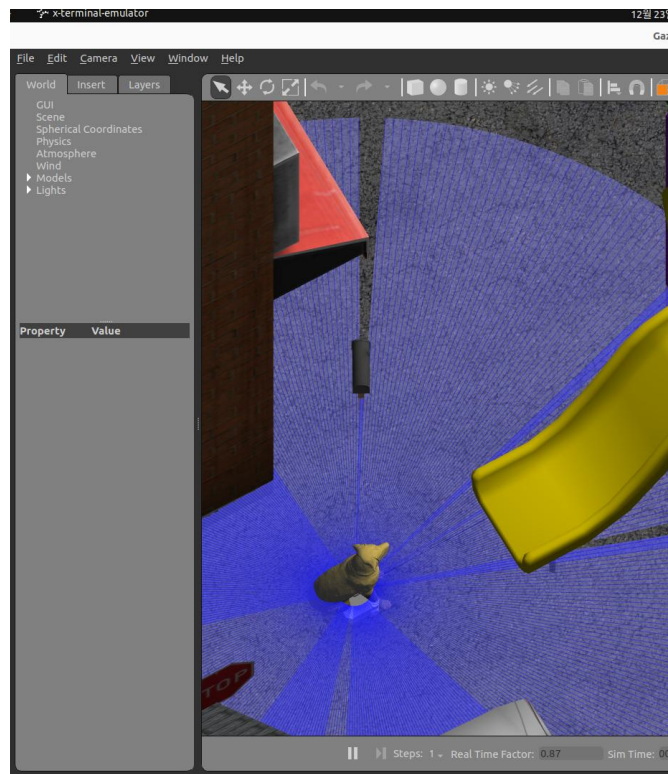
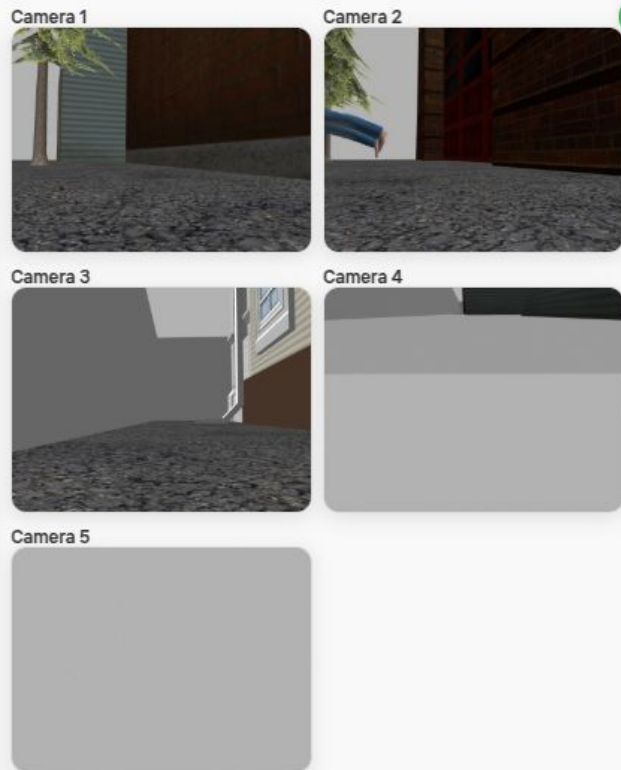


turtlebot3 waffle model 강아지로 개조 (땅에서 수색)



드론 새로 개조 (하늘에서 수색)

장착한 센서



터틀봇1,2,3: 카메라, LiDAR

드론: top, front(45도) 카메라

world 소개



평화롭던 마을...

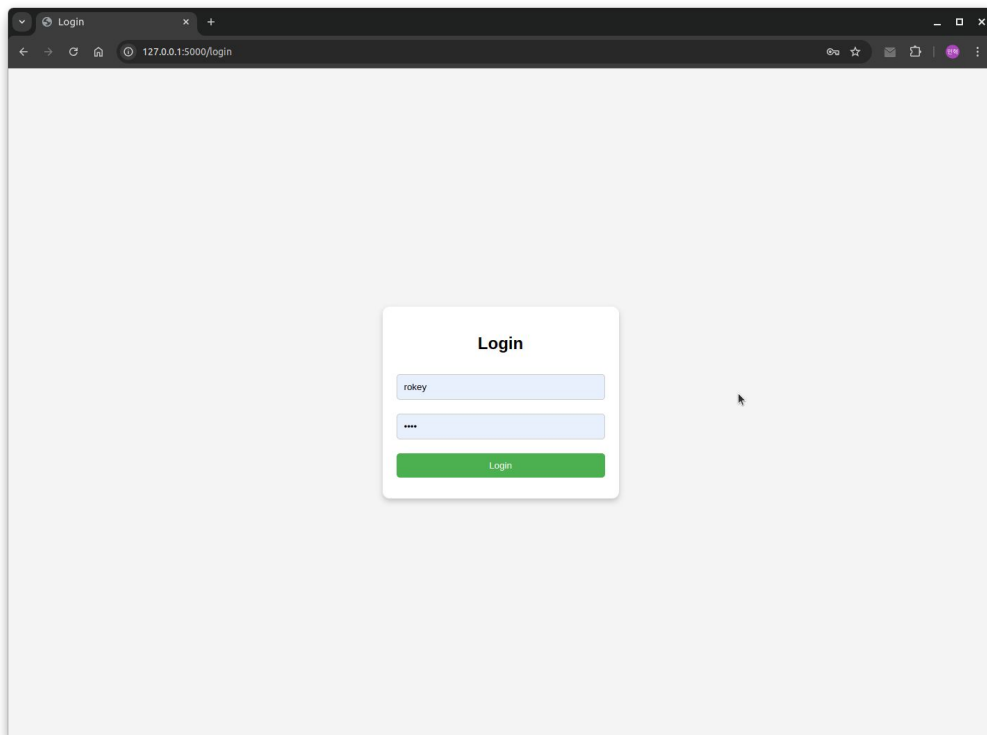
평 소리와 함께

재난이 시작되는데...

사람살려!!!

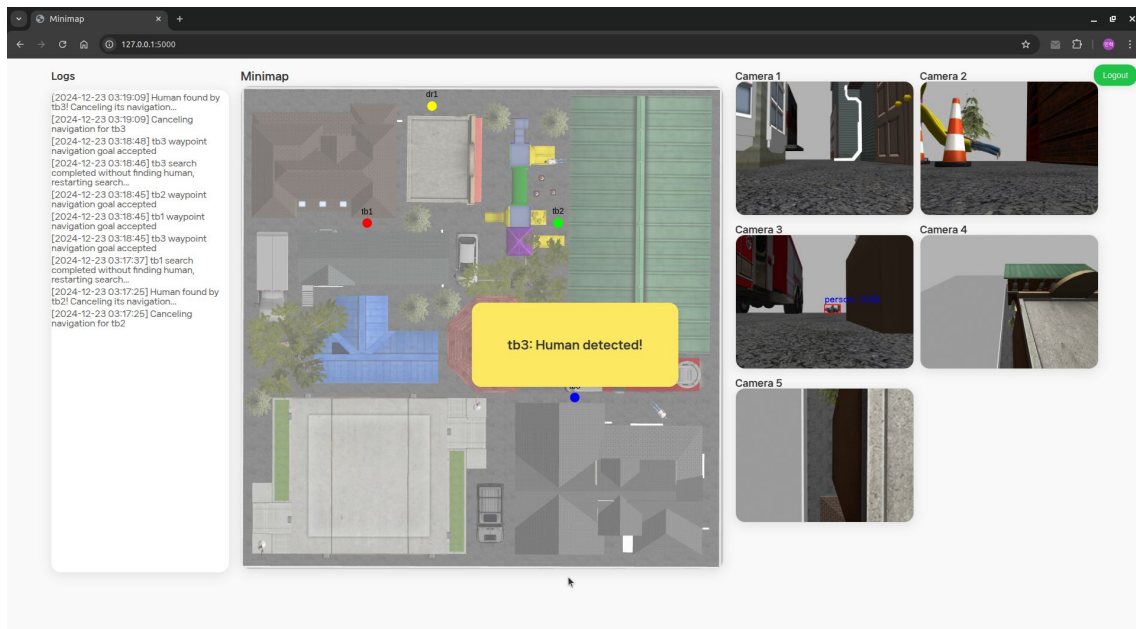
1. 라이다 범위 안에 두개 이상의 구조물이 인식되어 **SLAM**에 용이하도록 구성
2. 특징점을 주기 위해 나무, 자동차, 놀이터 등 불규칙적인 형태의 구조물 설치

관제시스템



Login: db와 연동되어, 암호화한
계정정보로 로그인을 통해
관리자만 화면에 접근이
가능하다

관제시스템



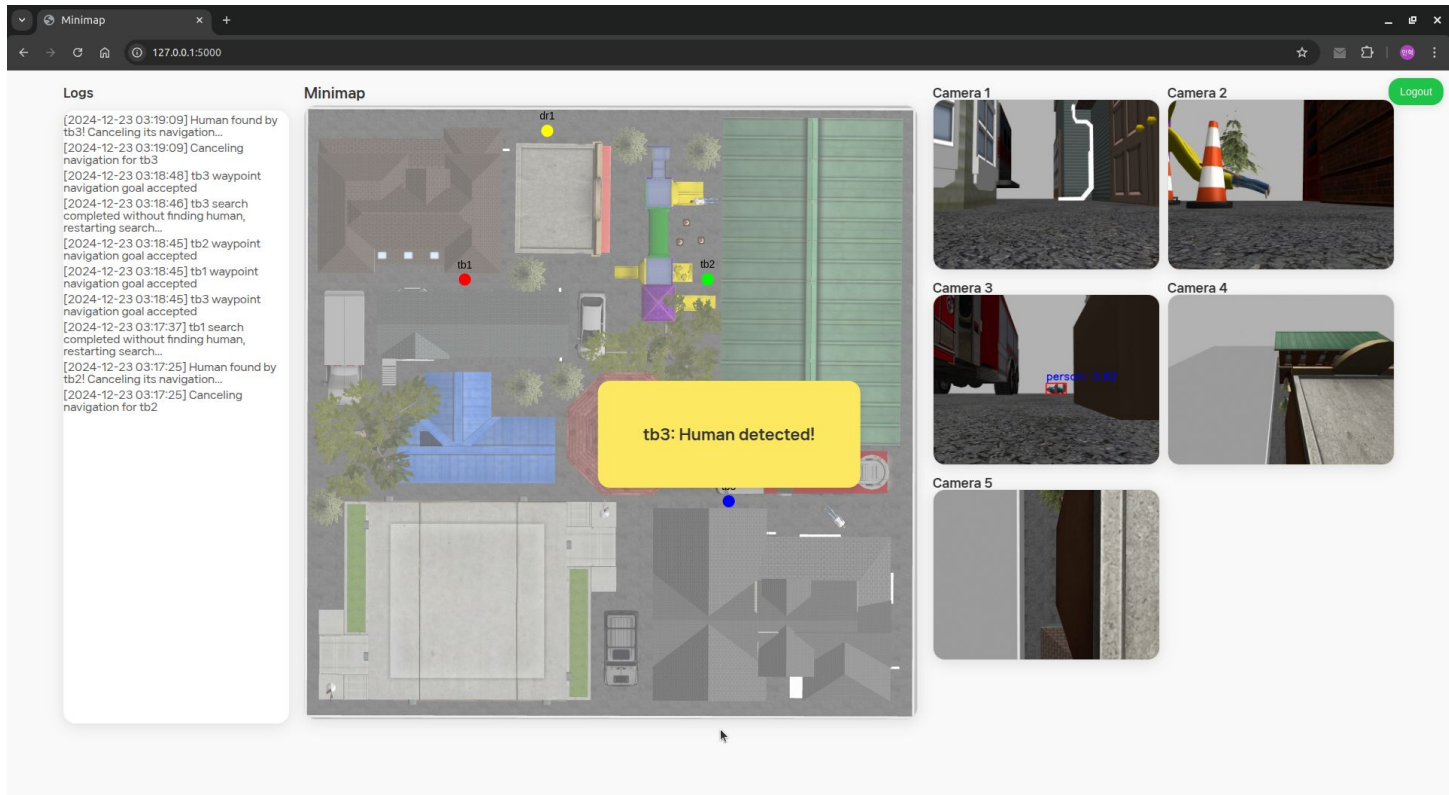
Popup: 사람을 발견하면 서버에서 응답을 받고, 팝업이 생성된다.

Camera: yolo가 적용된 각 로봇의 시야 화면을 **subscribe** 받아서 실시간으로 확인이 가능하다

Login: 로그인을 통해 관리자만 화면에 접근이 가능하다

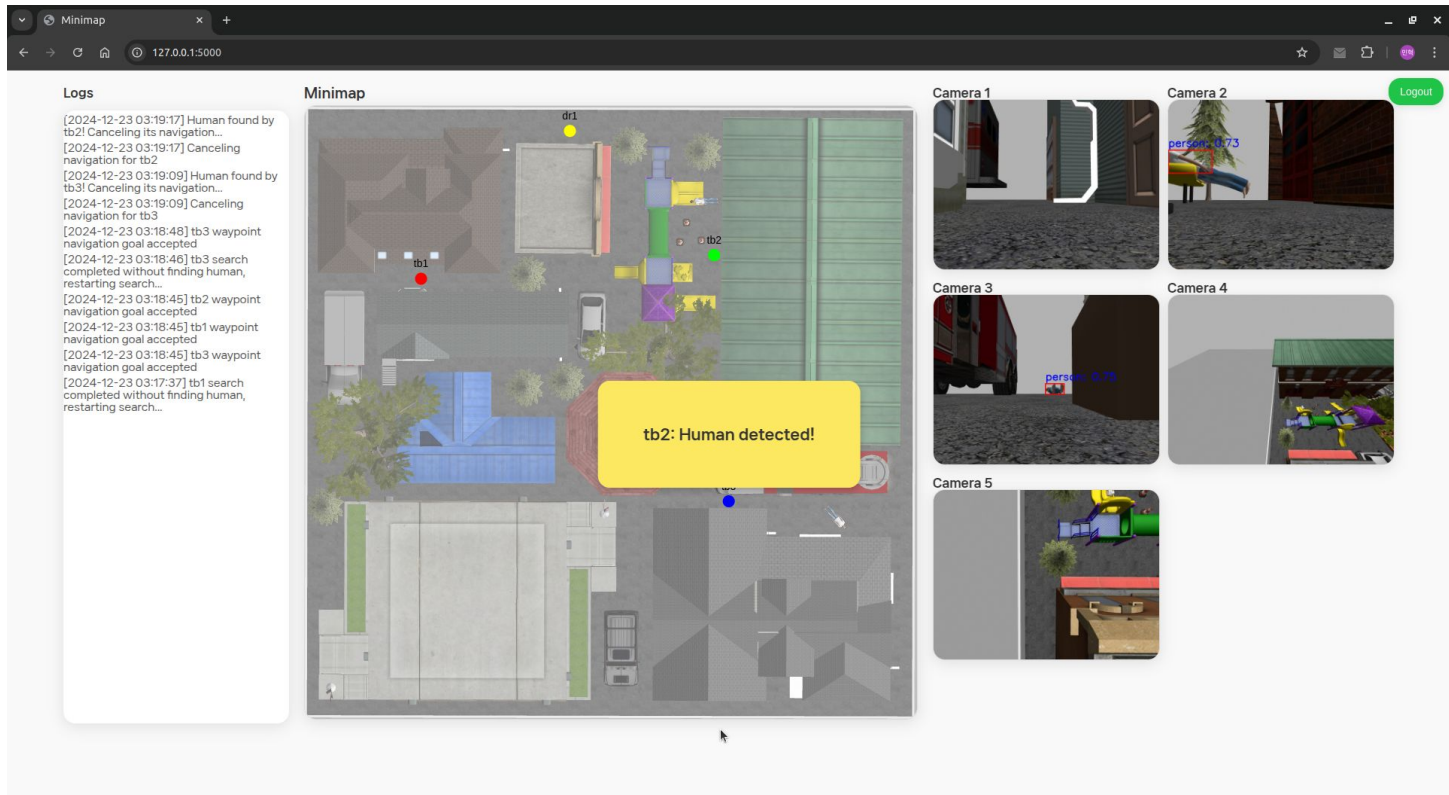
관제시스템

터틀봇3이 생존자 찾은
예시



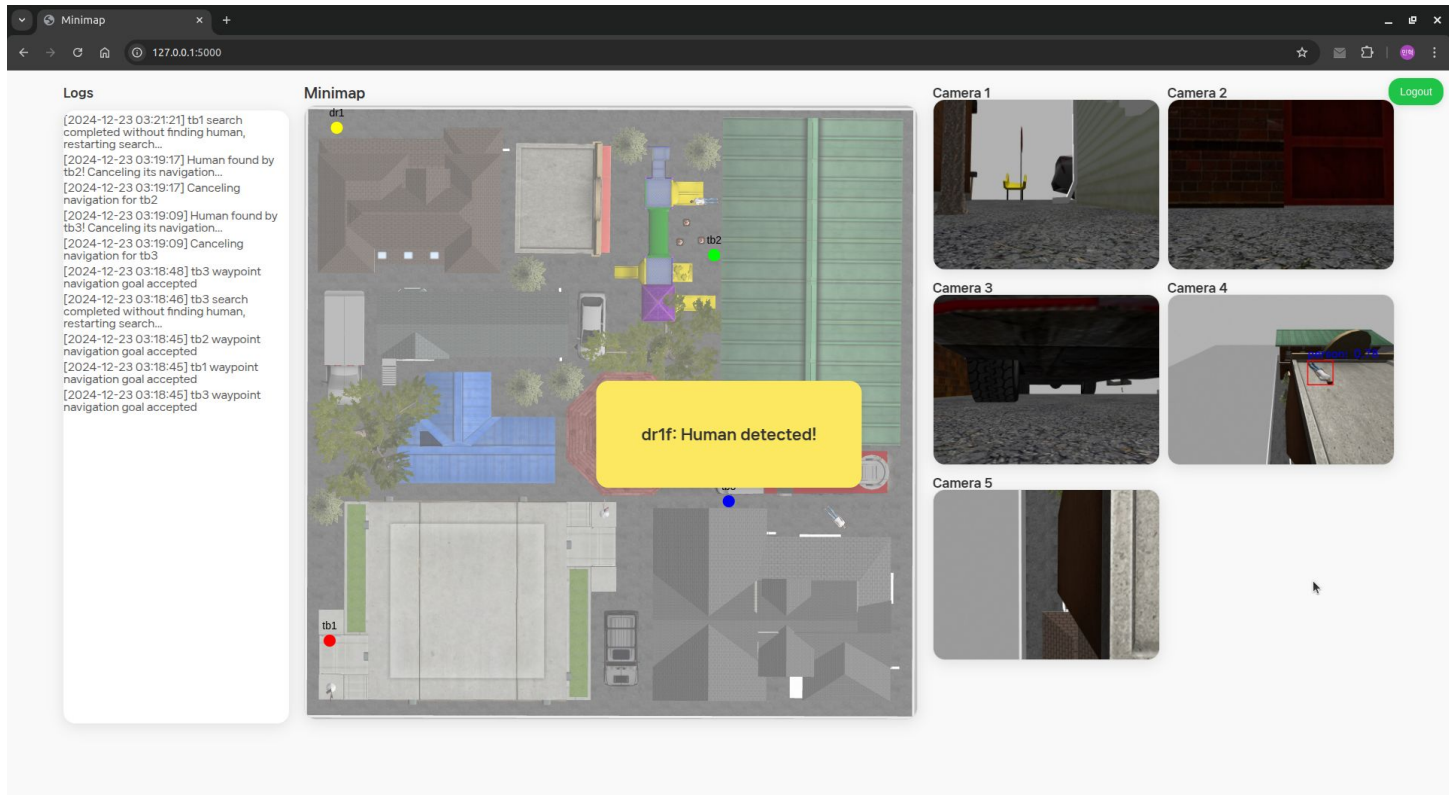
관제시스템

터틀봇2가 생존자 찾은
예시



관제시스템

드론의 front view 카메라가 생존자 찾은 예시



관제시스템 위치 표시 (좌표 변환)

속성

크기 814 × 814 픽셀

종류 PNG 그림

(-10,10)

Minimap



(30,-30)

Minimap : -)

터틀봇의 좌표를 받아와 맵의 실수 좌표(-10.0 ~ 30.0)를 이미지 상의 좌표(0~814)로 변환하는 과정을 거침

이 좌표를 db에 저장하고

구조현장의 지도 위에 db에서 실시간으로 로봇의 좌표를 받아와 위치가 업데이트되며 표시된다.

	x REAL	y REAL	name TEXT
1	531.9586249433205	469.58091455070974	tb3
2	534.6840094594577	289.17793781116114	tb2
3	171.73199919844993	231.8582807451557	tb1
4	40.725633452892986	30.529876276833978	dr1

관제시스템 log & db

Logs

```
[2024-12-23 03:09:29] tb3 waypoint navigation goal accepted
[2024-12-23 03:09:28] Human found by tb2! Canceling its
navigation...
[2024-12-23 03:09:28] Canceling navigation for tb2
[2024-12-23 03:09:27] tb3 search completed without finding
human, restarting search...
[2024-12-23 03:09:03] tb3 waypoint navigation goal accepted
[2024-12-23 03:09:01] tb3 search completed without finding
human, restarting search...
[2024-12-23 03:08:29] tb3 waypoint navigation goal accepted
[2024-12-23 03:08:27] tb3 search completed without finding
```

	x REAL	y REAL	name TEXT
1	531.9586249433205	469.58091455070974	tb3
2	534.6840094594577	289.17793781116114	tb2
3	171.73199919844993	231.8582807451557	tb1
4	40.725633452892986	30.529876276833978	dr1

```
message = f'{robot_id} waypoint navigation goal rejected'
```

```
self.log_to_db(message)
```

```
message = f'{robot_id} waypoint navigation goal accepted'
```

```
self.log_to_db(message)
```

```
message = f'{robot_id} search completed without finding human, restarting
search...'
```

```
self.log_to_db(message)
```

```
message = f'Human found by {robot_id}! Canceling its navigation...'
```

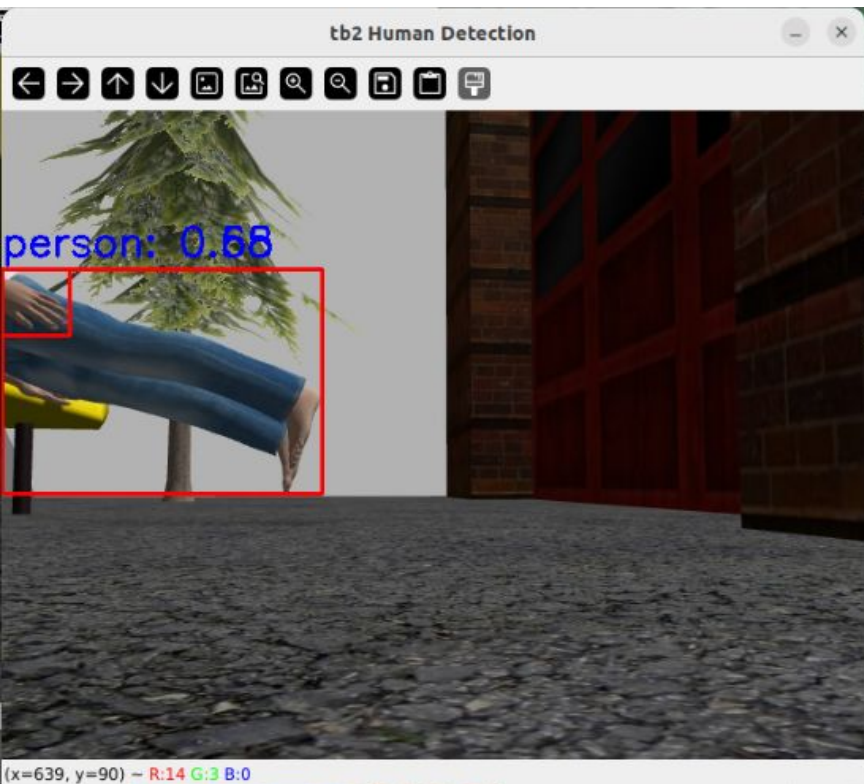
```
self.log_to_db(message)
```

```
message = f'Canceling navigation for {robot_id}'
```

```
self.log_to_db(message)
```

```
2024-12-23 04:29:17 tb3 waypoint navigation goal accepted
2024-12-23 04:29:17 tb2 waypoint navigation goal accepted
2024-12-23 04:29:17 tb1 waypoint navigation goal accepted
2024-12-23 04:29:22 tb1 search completed without finding human, restarting search...
2024-12-23 04:29:24 tb1 waypoint navigation goal accepted
2024-12-23 04:33:19 tb3 waypoint navigation goal accepted
2024-12-23 04:33:19 tb1 waypoint navigation goal accepted
2024-12-23 04:33:19 tb2 waypoint navigation goal accepted
2024-12-23 04:33:24 tb2 search completed without finding human, restarting search...
2024-12-23 04:33:26 tb2 waypoint navigation goal accepted
2024-12-23 04:33:46 Human found by tb3! Canceling its navigation...
2024-12-23 04:33:46 Canceling navigation for tb3
2024-12-23 04:34:19 Human found by tb2! Canceling its navigation...
2024-12-23 04:34:19 Canceling navigation for tb2
2024-12-23 04:35:16 tb1 search completed without finding human, restarting search...
```


yolo11s 모델 사용



모델	크기 (픽셀)	mAPval 50-95	속도 CPU ONNX (ms)	속도 T4 TensorRT10 (ms)	매개변수 (M)	FLOPs (B)
YOLO11n	640	39.5	56.1 ± 0.8	1.5 ± 0.0	2.6	6.5
YOLO11s	640	47.0	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
YOLO11m	640	51.5	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
YOLO11l	640	53.4	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
YOLO11x	640	54.7	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

yolov8 모델과 yolo11n, yolo11s, yolo11m, yolo11l 테스트 후,
응답 시간, 신뢰도를 고려하여 yolo11s로 결정

[multitb_interfaces] 서비스 인터페이스 패키지

- 각 로봇에 대한 팝업 서비스 존재
- 중복 팝업 방지를 위해 로봇 ID를 요청하고 성공 여부를 응답받는 개별 관리 프로그램
- 사람 감지 시, 팝업이 뜰 때 한 번만 뜨고 누르면 사라지게 하기 위해 표시 여부를 **False** 값으로 초기화한 상태에서 팝업 표시를 하면 **True**로 변환하여 응답 반환

이미 팝업이 표시된 상태라면 추가 요청 무시

```
if popup_displayed.get(robot_id, False):
```

```
    self.get_logger().info(
```

```
        f"Popup already displayed for {robot_id}. Ignoring request."
```

```
    )
```

```
    response.success = False
```

```
    return response
```

```
try:
```

```
    # 새로운 요청 처리
```

```
    self.message_queue.put_nowait(f"{robot_id}: Human detected!")
```

```
    popup_displayed[robot_id] = True # 팝업 표시 상태 업데이트
```

```
    self.get_logger().info(f"Popup displayed for {robot_id}.")
```

```
    response.success = True
```

```
GNU nano 6.2 HumanDetection.srv
# Request: 요청에 포함할 로봇 ID (tb1, tb2, tb3)
string robot_id
---
# Response: 응답으로 성공 여부 반환
bool success
```

[human_detector.py] ros2 통신

1. CompressedImage를 subscribe한 후, 사람 감지를 위한 yolo11s 모델 실행
2. 이미지 병렬 처리를 위해 큐를 활용한 멀티스레딩 구조 사용
3. 사람이 감지되면 내비 알림, 팝업 호출

```
# [Topic Publisher] yolo 감지 (내비용)
```

```
self.detection_pubs = {
```

```
    "tb1": self.create_publisher(String, "/tb1/human_detection", 10),
```

```
    "tb2": self.create_publisher(String, "/tb2/human_detection", 10),
```

```
    "tb3": self.create_publisher(String, "/tb3/human_detection", 10),
```

```
}
```

```
# [Service Client] 팝업 서비스
```

```
self.popup_clients = {
```

```
    "tb1": self.create_client(HumanDetection, "/tb1/popup_service"),
```

```
    "tb2": self.create_client(HumanDetection, "/tb2/popup_service"),
```

```
    "tb3": self.create_client(HumanDetection, "/tb3/popup_service"),
```

```
}
```

```
# [Topic Publisher] Flask에 publish
```

```
self.image_pubs = {
```

```
    "tb1": self.create_publisher(CompressedImage, "/tb1/processed_image/compressed", 10),
```

```
    "tb2": self.create_publisher(CompressedImage, "/tb2/processed_image/compressed", 10),
```

```
    "tb3": self.create_publisher(CompressedImage, "/tb3/processed_image/compressed", 10) }
```

사람 감지 시, 내비게이션에 알림

```
# yolo 감지 시, publish
if found_human:
    print("found human")
    msg = String()
    msg.data = "Human detected!"
    self.detection_pubs.publish(msg)

# 팝업 서비스 호출
self.call_popup_service()
```

사람 감지 시, 팝업 서비스 실행

yolo 처리한 이미지 flask에 전달

[turtlebot_navi.py] ros2 통신

```
# [Topic Subscriber] YOLO 감지 (내비용)
```

```
self.detection_subs = {  
    'tb1': self.create_subscription(String, '/tb1/human_detection',  
                                     lambda msg: self.human_detection_callback(msg, 'tb1'), 10),  
    'tb2': self.create_subscription(String, '/tb2/human_detection',  
                                     lambda msg: self.human_detection_callback(msg, 'tb2'), 10),  
    'tb3': self.create_subscription(String, '/tb3/human_detection',  
                                     lambda msg: self.human_detection_callback(msg, 'tb3'), 10)  
}
```

```
# [Action Client] 각 터틀봇 Waypoint 내비
```

```
self.waypoint_clients = {  
    "tb1": ActionClient(self, FollowWaypoints, "/tb1/follow_waypoints"),  
    "tb2": ActionClient(self, FollowWaypoints, "/tb2/follow_waypoints"),  
    "tb3": ActionClient(self, FollowWaypoints, "/tb3/follow_waypoints"),  
}
```

```
# goal handle이 있는 경우에만 취소  
if robot_id in self.goal_handles:  
    goal_handle = self.goal_handles[robot_id]  
    if goal_handle:  
        message = f'Canceling navigation for {robot_id}'  
        self.get_logger().info(message)  
        self.log_to_db(message)  
        goal_handle.cancel_goal_async()
```

사람 감지 토픽 받으면
콜백함수에서 실행 중인 action
취소 처리

각 로봇에 대해 waypoints 기반
경로 순차

[turtlebot_navi.py] ros2 통신

1. launch 파일로 spawn 된 각 로봇에 대해 순차 경로 지정
2. navigation action
3. /amcl_pose의 pose 값을 subscribe 하고 좌표변환하여 db에 robot_id와 함께 저장
4. 각 로봇의 상태 관리 (사람을 찾았는지, 탐색할건지)

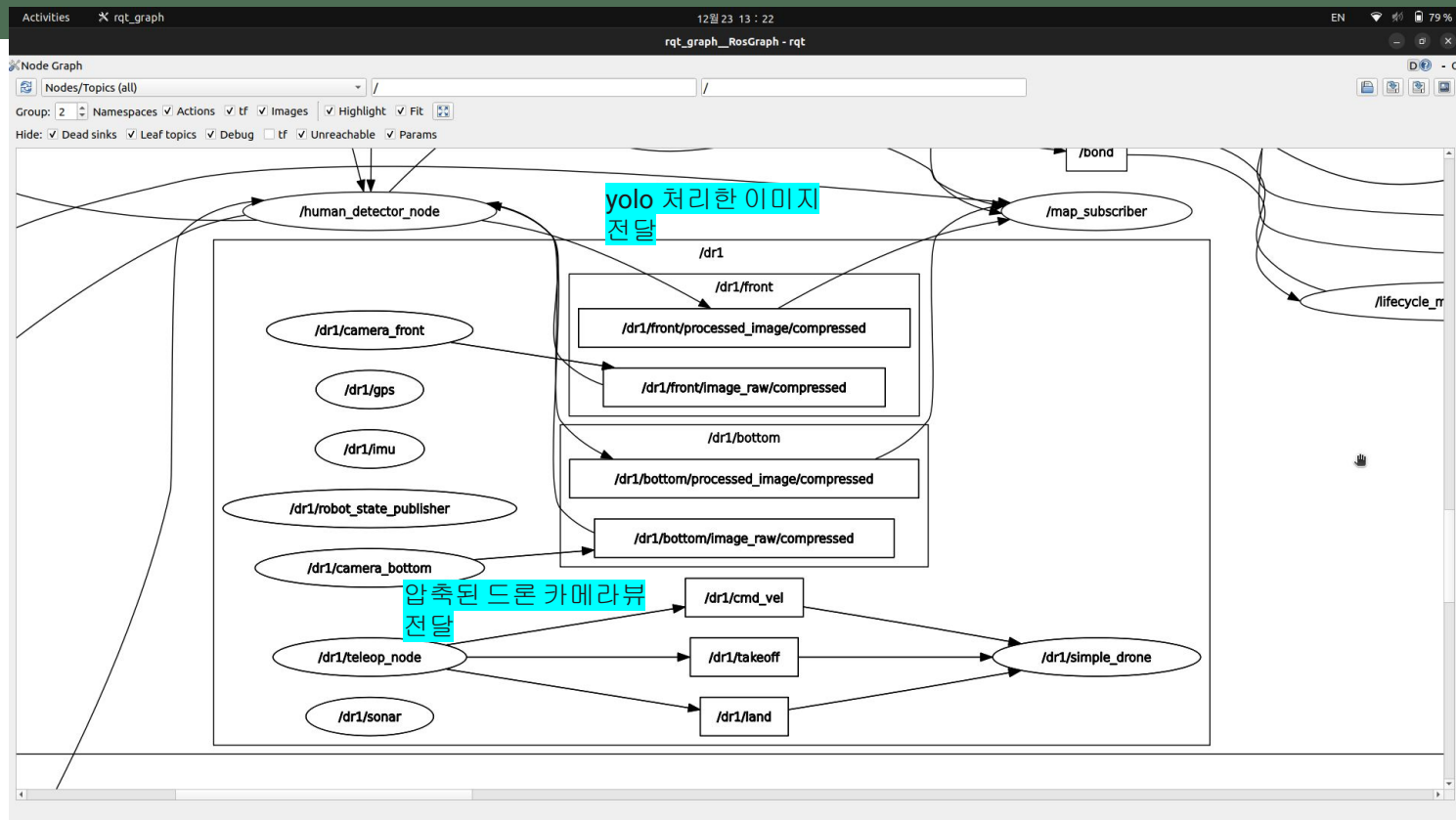
각 터틀봇의 웨이포인트 설정

```
self.waypoints = {
    'tb1': [
        {'position': [9.295630986849401, -0.5544288793639277], 'orientation': [0.15878652944779836, 0.9873129382652308]},
        {'position': [10.365256866305076, 7.787880983302672], 'orientation': [0.8404702519697483, 0.5418576893926187]},
        {'position': [-0.4387401370420988, 9.071016132538075], 'orientation': [0.9977366712317057, 0.06724235926464121]},
        {'position': [-9.45115053678013, 6.897161746293658], 'orientation': [-0.7674129621733148, 0.6411531373146189]},
        {'position': [2.0232847747817924, -1.3291683400306933], 'orientation': [0.058881284356604745, 0.998264992049965]}],
    'tb2': [
        {'position': [16.165101740614062, -5.747140591243882], 'orientation': [-0.590045409570518, 0.8073700605328139]},
        {'position': [16.976360093201357, 0.7948136035872039], 'orientation': [0.8466122734932133, 0.5322101637235545]},
        {'position': [27.889480573673328, 9.27738469393071], 'orientation': [0.05753832844685656, 0.9983432980490937]},
        {'position': [29.11476630716706, -11.129364021192913], 'orientation': [-0.7146300809002273, 0.6995025714552696]},
        {'position': [22.516774684472146, -12.023316619940662], 'orientation': [-0.9995251252549825, 0.030814347048274782]}],
    'tb3': [
        {'position': [16.09692885803449, -13.0612694406839], 'orientation': [-0.6372530217208481, 0.7706546478855807]},
        {'position': [17.9209957051445, -15.940316654694865], 'orientation': [-0.19407855179166453, 0.9809859916096917]},
        {'position': [22.2851252302164, -16.259286435777494], 'orientation': [-0.10289096948012709, 0.9946926401655134]},
        {'position': [24.518450791571095, -15.854008904372686], 'orientation': [-0.03573459755819703, 0.99936131530961]},
        {'position': [29.154423581940755, -24.232828661615255], 'orientation': [-0.6017866747370478, 0.7986568713214246]},
        {'position': [29.490570904874442, -26.656547508314297], 'orientation': [-0.5453998575784409, 0.8381759930667404]},
        {'position': [12.875300774942577, -29.3808703571920], 'orientation': [0.9998315225954153, 0.01835555568577111]},
        {'position': [12.046465734617959, -21.43655296589399], 'orientation': [0.800983083053206, 0.5986869805353887]}]
```

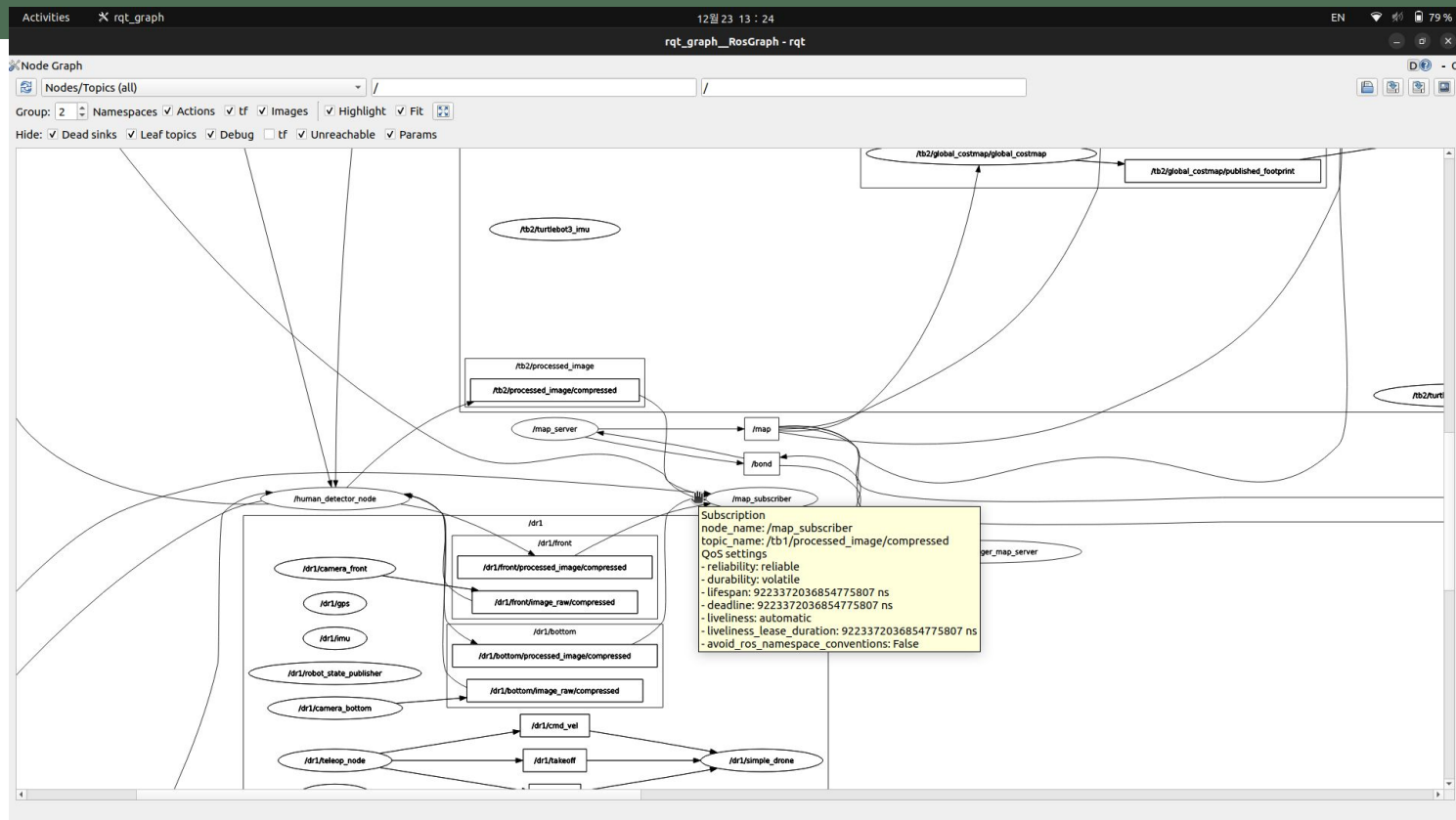
각 터틀봇의 상태 관리

```
self.robot_states = {
    'tb1': {'searching': False, 'human_found': False},
    'tb2': {'searching': False, 'human_found': False},
    'tb3': {'searching': False, 'human_found': False}
}
```

rqt



rqt



파라미터 설정

nav2_params.yaml

update_min_a: 0.05 # 각

update_min_d: 0.05 # 위치 업데이트에 필요한 최소 이동 거리

width: 30 #SLAM에 유지할 윈도우의 크기

height: 30

시연 영상

Activities x-terminal-emulator 12월 23 13 : 33 0:00 EN 79%

Minimap 127.0.0.1:5000

Logs

- [2024-12-23 04:29:24] tb1 waypoint navigation goal accepted
- [2024-12-23 04:29:22] tb1 search completed without finding human, restarting search...
- [2024-12-23 04:29:17] tb3 waypoint navigation goal accepted
- [2024-12-23 04:29:17] tb2 waypoint navigation goal accepted
- [2024-12-23 04:29:17] tb1 waypoint navigation goal accepted
- [2024-12-23 04:18:35] Human found by tb3! Canceling its navigation...
- [2024-12-23 04:18:35] Canceling navigation for tb3
- [2024-12-23 04:17:51] tb2 waypoint navigation goal accepted
- [2024-12-23 04:17:51] tb1 waypoint navigation goal accepted
- [2024-12-23 04:17:51] tb3 waypoint navigation goal accepted

Minimap

Camera 1

Camera 2

Camera 3

Camera 4

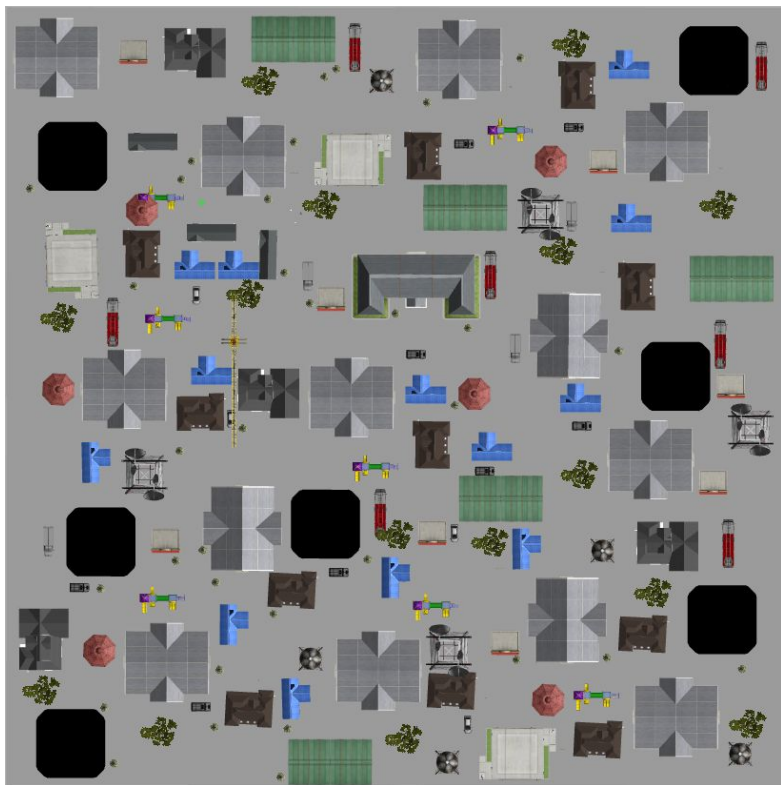
Camera 5

Real Time: 00:00:00.47.991 Iterations: 2329 FPS: 62.50

```
Idingg@idingg-laptop: ~/multitb_ws
colcon build [7/7 done] [0 ongoing] Z
...
[ros2-8] Waiting for at least 1 matching subscription(s)
...
Idingg@idingg-laptop: ~/multitb_ws
127.0.0.1 - [23/Dec/2024 13:33:16] "GET /data HTTP/1.1" 200 -
Idingg@idingg-laptop: ~/multitb_ws [dri] Received image for processing
Idingg@idingg-laptop: ~/multitb_ws
(base) Idingg@idingg-laptop: ~/multitb_ws$ source install/setup.bash && ros2 run yolo_human_navi turtlebot_navi
```


개선 사항

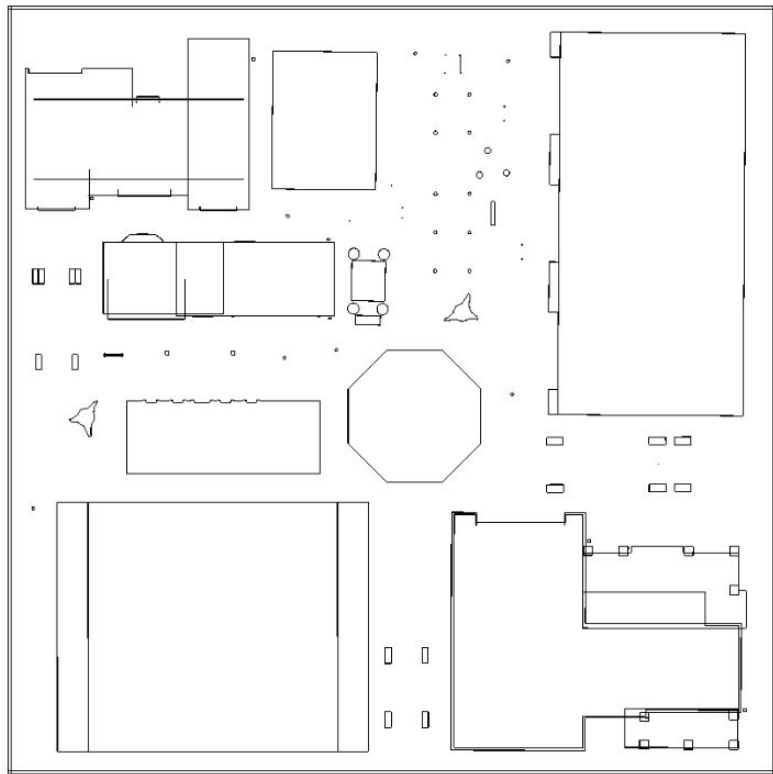
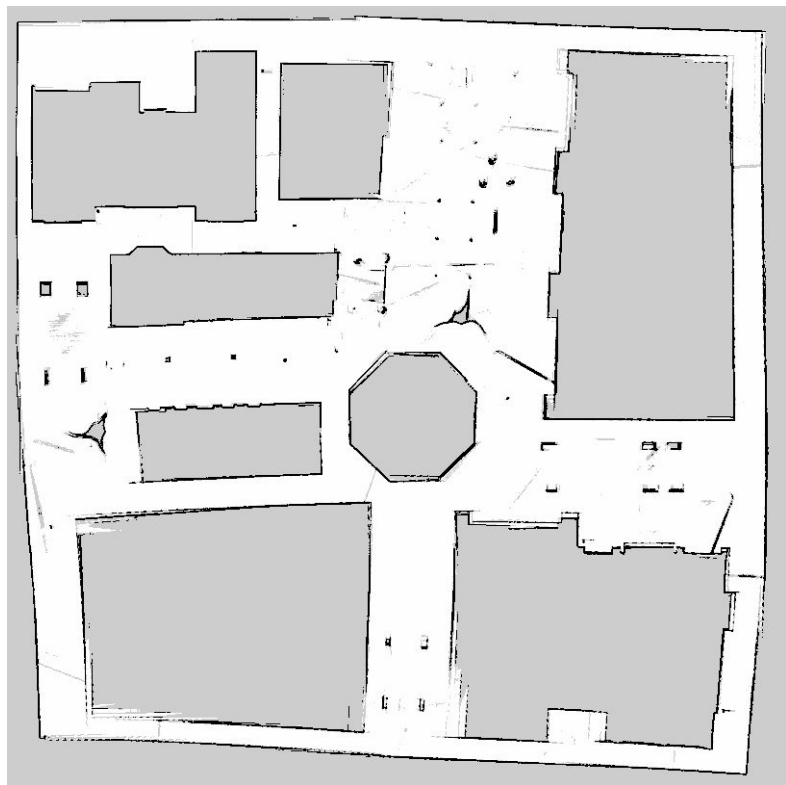
slam 한 영역당 15분 총 25시간 소요될 것으로 예상되어
world 크기 1/25로 줄임



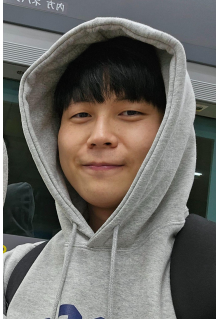
개선 사항

SLAM이 틀어지는 문제 발생 -> .world의 데이터로 맵 생성

https://github.com/arshadlab/gazebo_map_creator



Team Contribution



진민혁
드론



서연우
터틀봇



배민지
관제시스템

감사합니다.

و (٩)

