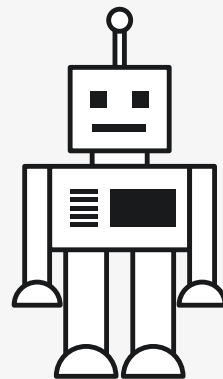
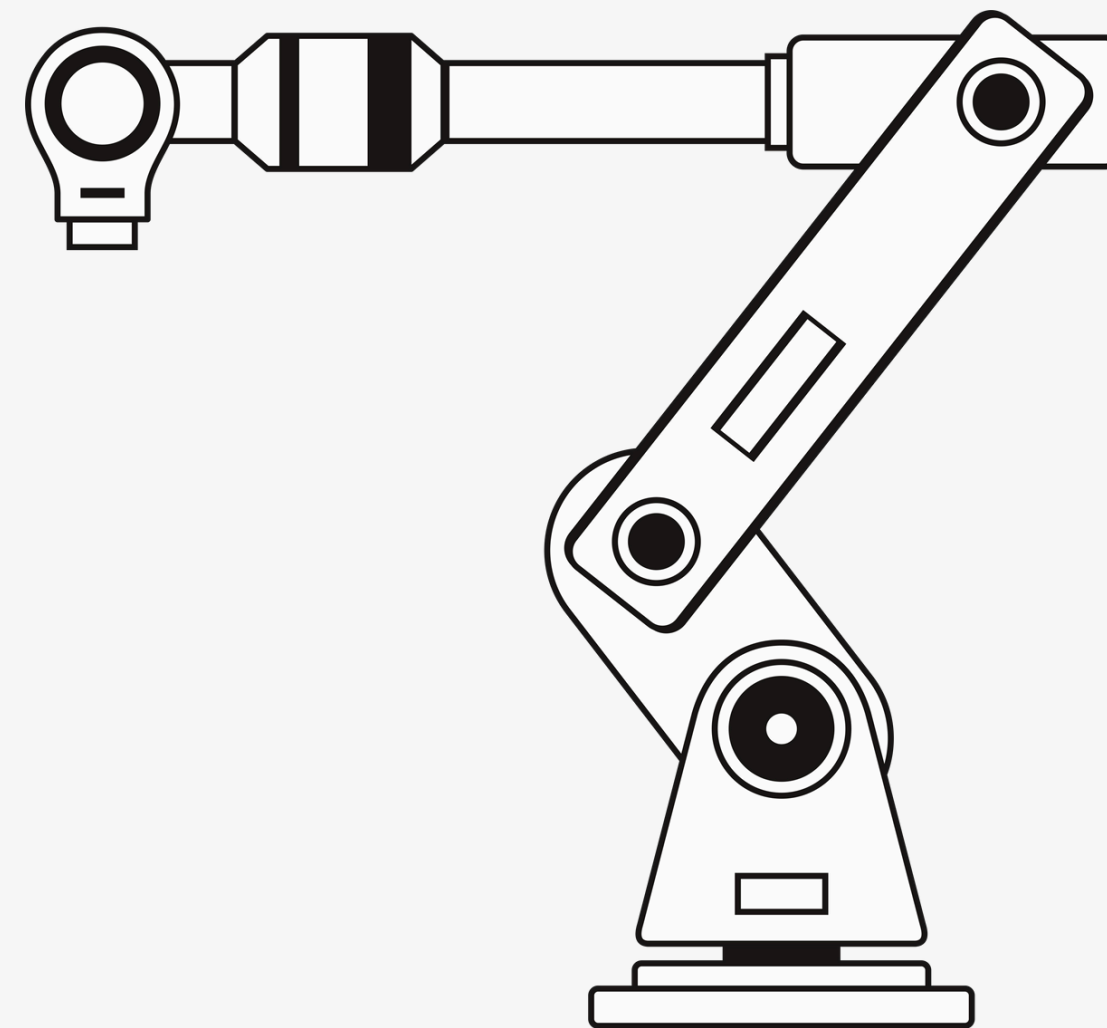


스마트 시대를 여는 오피스 서비스 혁신 전략



디지털 트윈 기반 로봇 자동화 시뮬레이션 시스템 구현 (김민수 수석)



F3조 _공조

강동혁 | 김갑민 | 김효원 | 황수빈 | 황혜인

Contents

01 프로젝트 개요

02 프로젝트 팀 구성 및 역할

03 프로젝트 수행 절차 및 방법

04 프로젝트 개발 목차

05 프로젝트 수행 경과

06 프로젝트 결과 및 분석

07 자체 평가 의견

08 질문 및 답변

01 프로젝트 개요

주제 및 선정배경

오피스 내부 물품
이동 자동화 필요성

실내 자율주행 기술
학습 및 구현 목적

프로젝트 내용

오피스 환경 제작
+ Map 생성

Nav2 기반 자율주행
(목적지 지정 → 이동)

자율 주행
로봇 사용

목적지 도착 시 박스
부착 이벤트 구현

활용 기술

Isaac Sim

Ros2 + Nav2

Python Script

Rviz2

프로젝트 구조

환경 구성
→ Map 생성

Nav2 연동
→ 목적지 설정

로봇 자율주행 수행

박스 인식
→ 자동 부착

기대 효과

디지털 트윈 및 Isaac
Sim 기반 로봇 시뮬
레이션 역량 확보

Nav2 및 Ros2 기반
자율주행 시스템에
대한 실전 경험 축적

Isaac Sim 환경에서
의 객체 상호작용
매커니즘 이해

새로운 3D 환경 구축
및 디지털 트윈 모델
링 경험

02 프로젝트 팀 구성 및 역할

조원	역할	담당 업무
강동혁	팀장	Officerroom Design 자율주행로봇 load 및 Ros2 Bridge 활성화
김갑민	팀원	Custom Map에 맞춰 자율주행로봇 최적화 Table과 Box 간 Attach 기능 구현 및 디버깅
김효원	팀원	Custom Map의 Occupancy Map 생성 Box와 자율주행로봇 간 Attach 및 Reparent 기능 구현
황수빈	팀원	Custom Occupancy Map 최적화 및 서버 등록 Box와 자율주행 로봇 간 Attach 기능 구현
황혜인	팀원	Custom Officerroom USD file 생성 시뮬레이션 환경 세팅 및 Rviz를 통한 map 주행 검증
멘토 : 이현종, 엄정민, 박재형 연구원		프로젝트 기술 자문

03 프로젝트 수행 절차 및 방법

구분	기간	활동	비고
기술 학습	11/10(월) ~ 11/14(금)	Isaac Sim 기본 사용법 학습 오피스 환경 생성, Prim 구조 이해	공통 기술 숙지 단계
사전 기획	11/14(금) ~ 11/16(일)	최종 주제 선정 환경 구성 방향 및 역할 분담 확정	개발 필요 기능 명세 정리
환경 제작 및 맵 생성	11/17(월) ~ 11/18(화)	오피스 구조 모델링 Occupancy Map 생성	경로계획 테스트 준비
Nav2 연동 및 자율주행 구현	11/18(화) ~ 11/19(수)	Rviz2 목표지 설정 테스트 자율주행 기능 검증	이동 안정화 작업
박스 자동 부착 기능 구현	11/19(수) ~ 11/20(목)	로봇 - 박스 거리 계산 코드 구현 박스 유지 및 전달 기능 점검	Python Script 기반 개발
최종 점검 및 발표	11/21(금)	전체 기능 통합 테스트 오류 수정 및 시나리오 정리	프로젝트 최종 발표

04 프로젝트 개발 목차

1

Office 환경 구축

2

Occupancy Map
설정

3

Nav2 Bring up
+ 자율주행 실행

4

로봇과 Cube 월드
좌표 실시간 계산

5

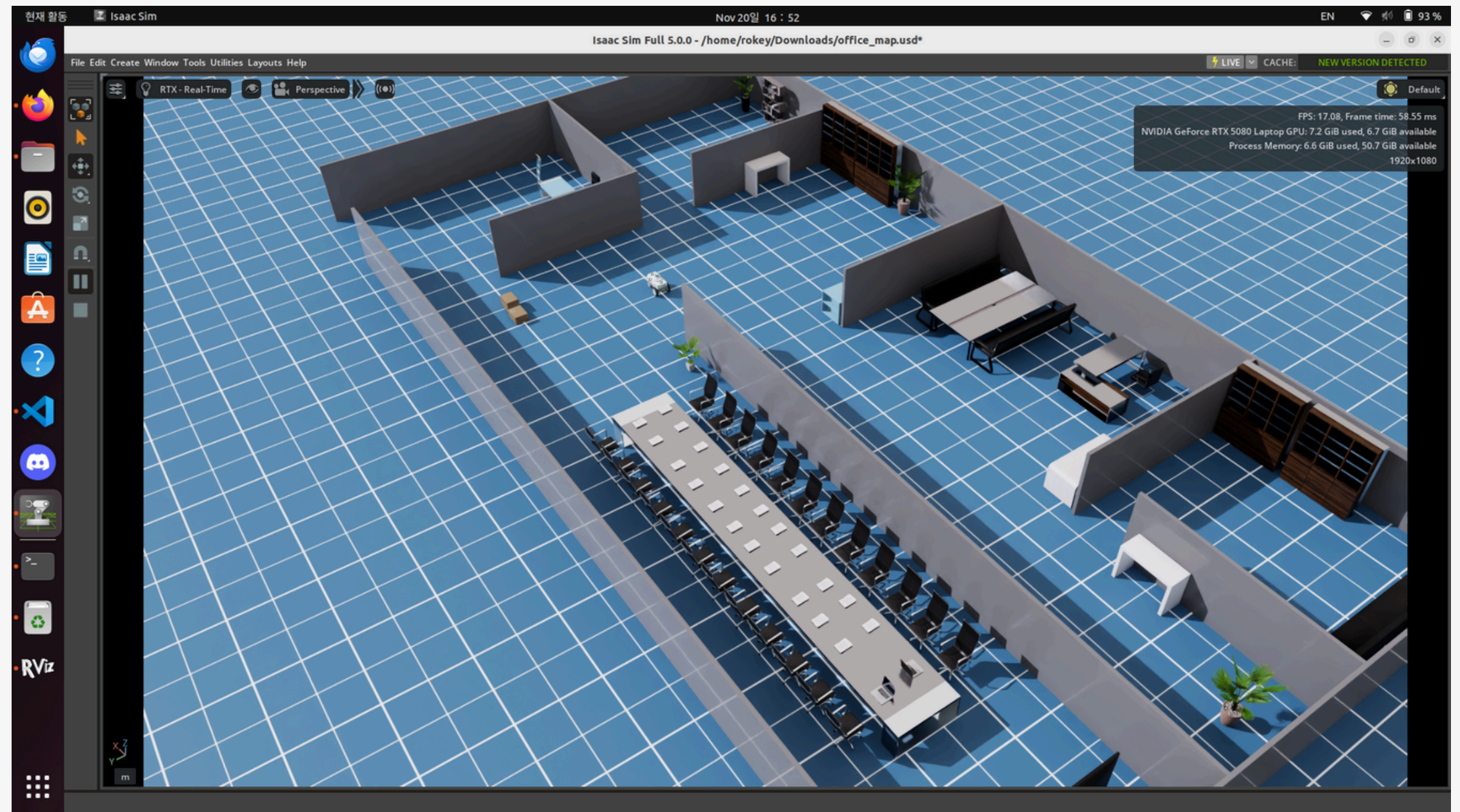
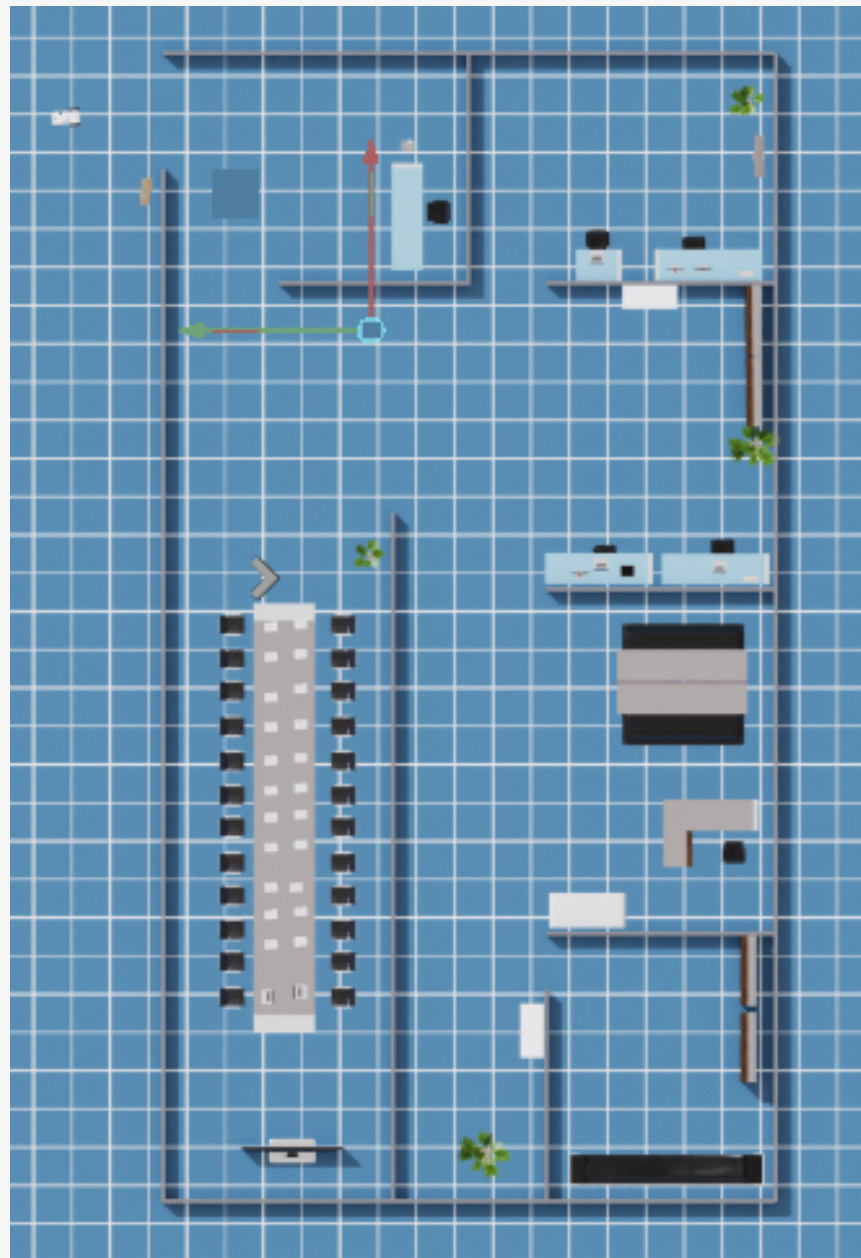
거리 조건 만족 시 자동
부착 (Re-parent)

6

부착 후 다음 Nav2
goal로 이동

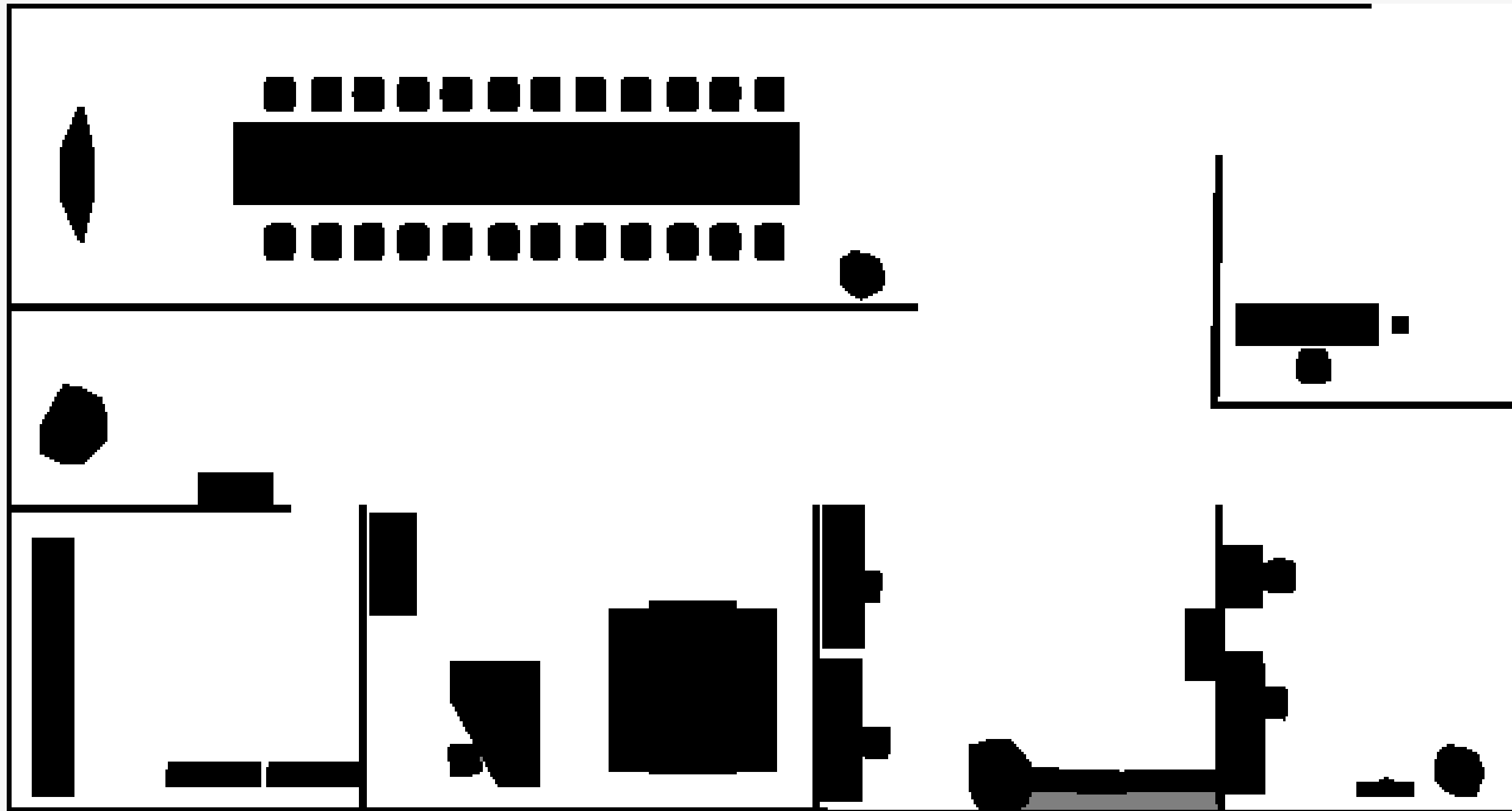
05 프로젝트 수행 경과

(1) isaac sim에서 Office 환경 구축



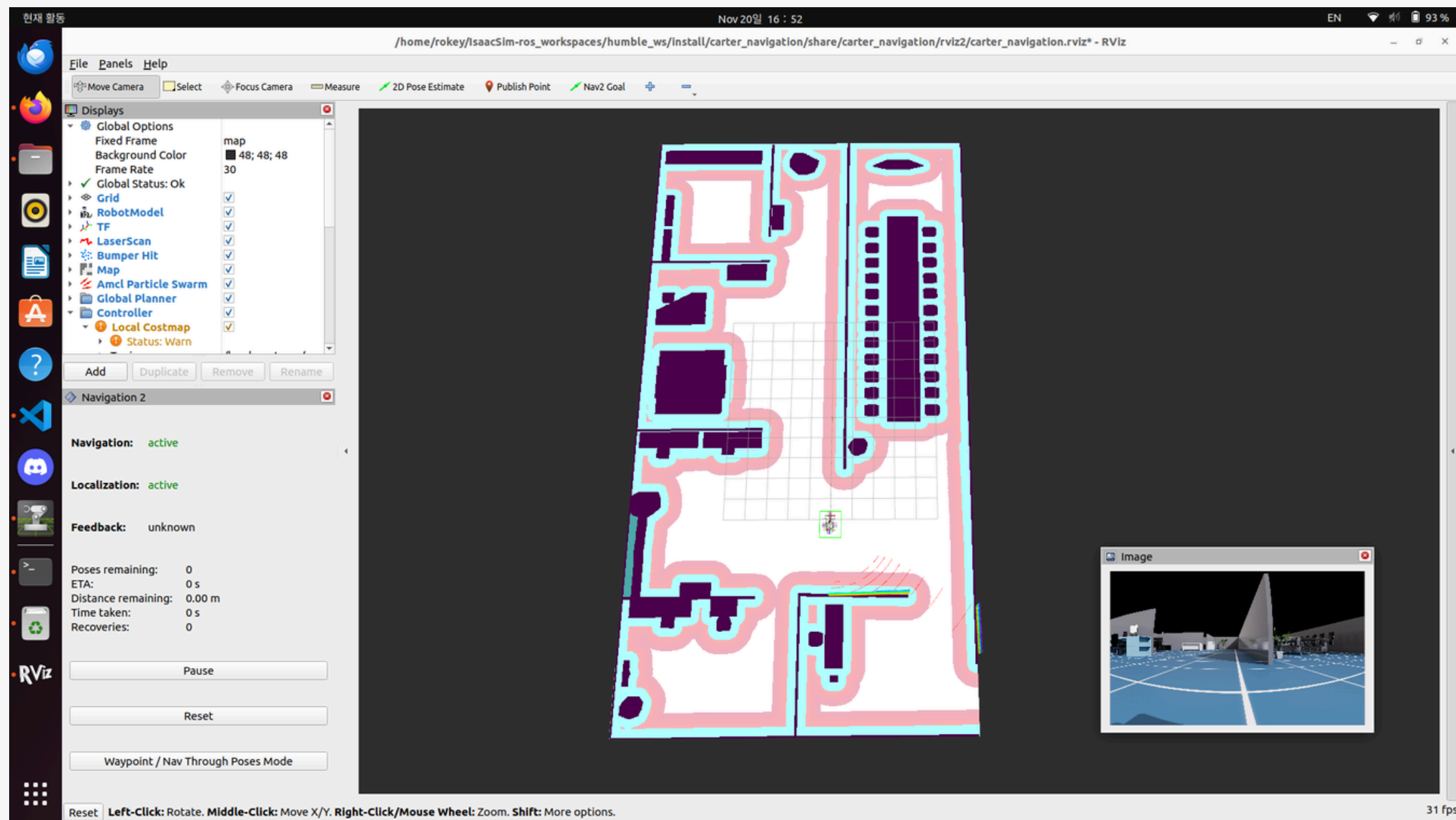
05 프로젝트 수행 경과

(2) Occupancy Map 설정



05 프로젝트 수행 경과

(3) Nav2 Bring up + 자율주행 실행



Nav2 및 RViz 기반 자율주행 시뮬레이션 화면

- 맵 상에서 로봇의 위치, 경로, 장애물 인식 정보가 실시간 표시됩니다.
- Navigation2를 실행하면, 지정된 목적지 까지 로봇이 스스로 경로를 찾아 이동할 수 있습니다.

05 프로젝트 수행 경과

(4) 로봇과 Box1,2의 World 좌표 실시간 계산

```
def on_update(dt):
    global is_attached
    global target_01_state
    global target_02_state

    # 로봇, 상자, 목적지 위치 데이터 정보
    robot_pos = get_world_position(BASE_LINK_PATH)
    box_01_pos = get_world_position(BOX_01)
    box_02_pos = get_world_position(BOX_02)
    target_01_pos = get_world_position(TARGET_01)
    target_02_pos = get_world_position(TARGET_02)
```

→ 로봇, Box 위치 데이터 얻기

```
if not target_01_state and target_01_dist < limit_distance:
    stage.RemovePrim(BOX_01_JOINT_PATH)

    input0 = Sdf.Path(TARGET_01)
    input1 = Sdf.Path(BOX_01)
    joint_prim = UsdPhysics.FixedJoint.Define(stage, TARGET_01_JOINT_PATH)
    joint_prim.CreateBody0Rel().SetTargets([input0])
    joint_prim.CreateBody1Rel().SetTargets([input1])

    is_attached = False
    target_01_state = True

    print("📦 box1 complete!!")
```

→ 목표 지점 도착 시 Place

05 프로젝트 수행 경과

(4) 로봇과 Box1,2의 World 좌표 실시간 계산

```
if not target_01_state and box_01_dist < limit_distance:
    input0 = Sdf.Path(BASE_LINK_PATH)
    input1 = Sdf.Path(BOX_01)

    joint_prim = UsdPhysics.FixedJoint.Define(stage, BOX_01_JOINT_PATH)
    joint_prim.CreateBody0Rel().SetTargets([input0])
    joint_prim.CreateBody1Rel().SetTargets([input1])
    joint_prim.CreateLocalPos0Attr().Set(Gf.Vec3f(-0.2, 0.0, 1.0))

    print("📦 box1 attached!!")
    is_attached = True
```

→ Box Pick

```
subscription = omni.kit.app.get_app().get_update_event_stream().create_subscription_to_pop(on_update)
```

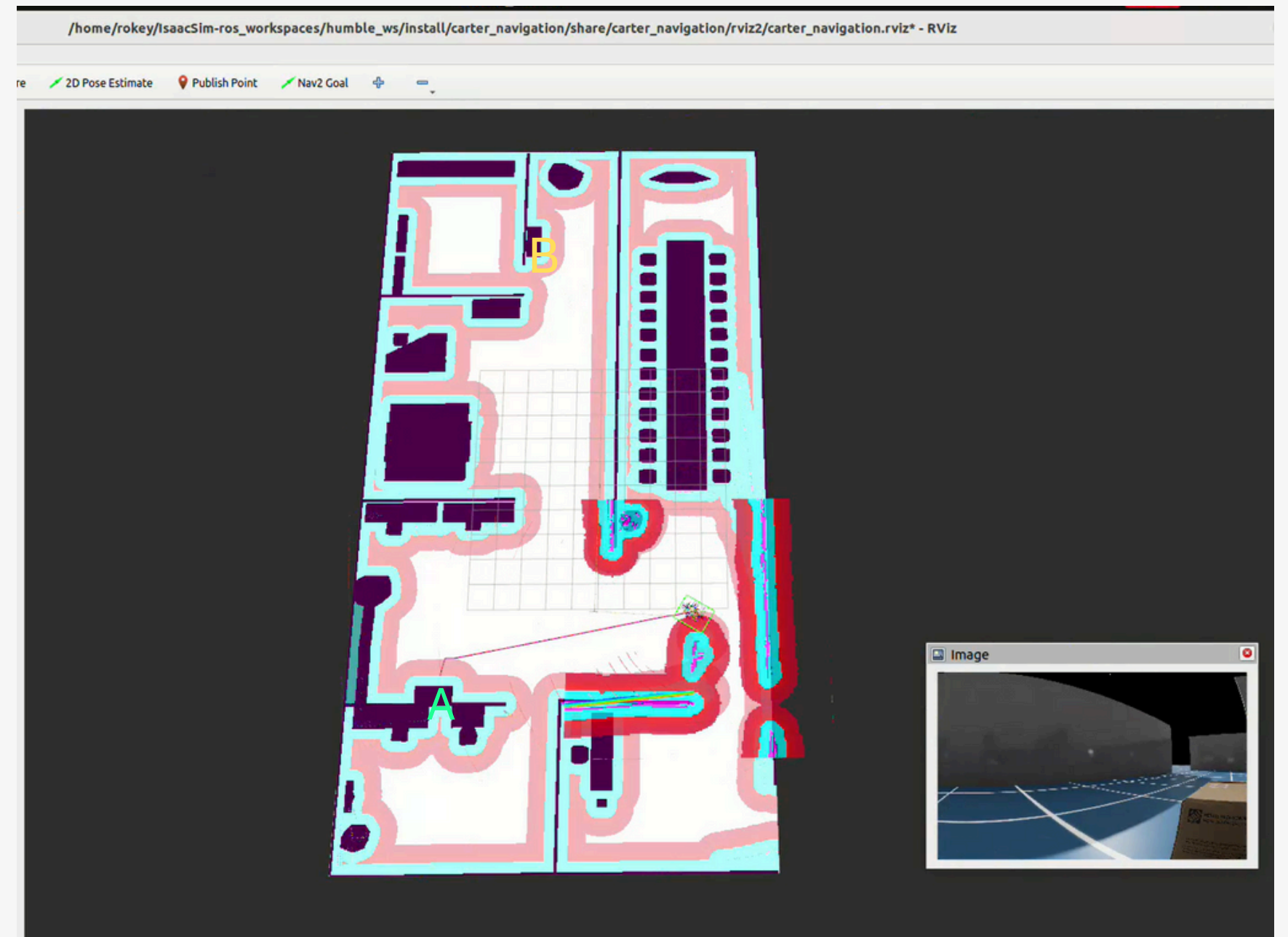
→ 업데이트 콜백 등록

05 프로젝트 수행 경과

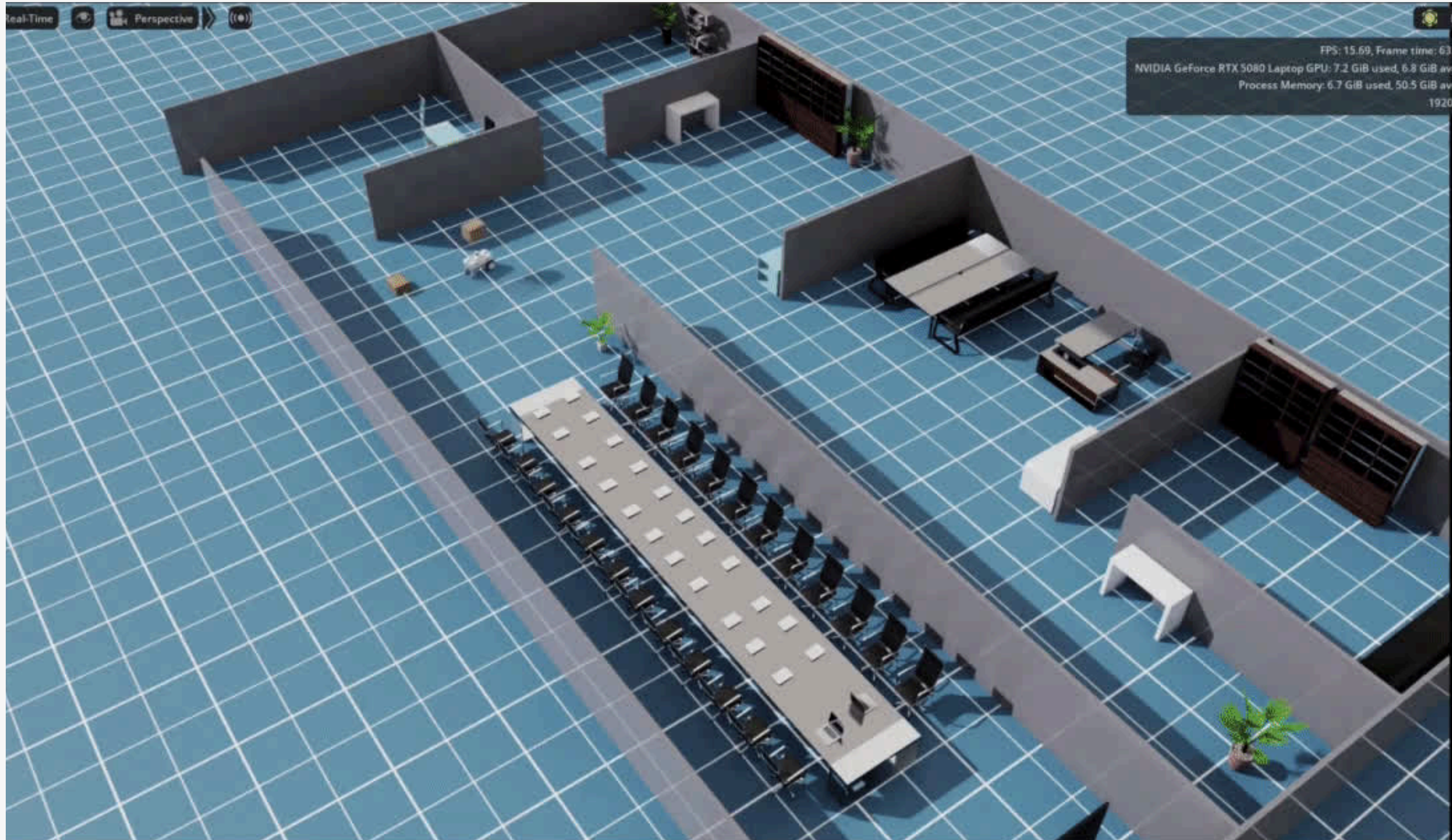
(5) 거리 조건 만족 시 자동 부착



(6) 부착 후 다음 Nav2 goal로 이동



06 프로젝트 결과 및 분석



07 자체 평가 의견

완성도 평가

9 / 10

- 목표였던 '사무실 환경에서의 자율주행 및 배송 기능'을 성공적으로 구현함
- Nav2 기반 경로 탐색, 목적지 지정 이동, 박스 자동 부착 기능까지 정상적으로 동작함

우리 팀이 잘한 부분 / 아쉬운 점

< 잘한 부분 >

- 피드백 반영 및 팀 내 의사소통이 원활하게 이루어짐
- 문제 상황 발생 시 빠른 대처와 수정이 이루어짐

< 아쉬운 점 >

- 레퍼런스 부족으로 구현 방향을 잡는 데 많은 시간이 소요됨

추후 개선점 및 보완할 점

- 해당 프로젝트의 진행 기간이 더 길었다면, 단순한 박스의 이동이 아닌 Gripper를 통하여 Pick & Place과정까지 구현할 수 있음
- 우리팀은 Jakal과 Gripper의 연결까지 성공하였기때문에 충분히 구현가능하다고 판단됨

느낀 점 및 경험한 성과

- Isaac Sim과 ROS2를 실제로 연동하면서 시뮬레이션 기반 로봇 개발 프로세스를 실전처럼 경험함
- Nav2 경로 계획의 구조와 파이프라인을 직접 다루며 자율주행 알고리즘의 전체 흐름을 체득함

질문 및 답변

궁금한 사항을 질문해 주세요.

Thank you

감사합니다.