GOOD MORNING!
早上好!
안녕하세요!

DAY 3

DAY2 RECAP

DAY 1

- Welcome
- Project Introduction
- Introduction to Project Development Process
- Business Requirement Development
- System Requirement Development
- System and Development environment Setup

DAY 2 (MINI PROJECT)

- Yolo객체 인식 모델 활용과 성능 평가 방법 이해
  - Custom Dataset과 Fine Tuning으로 자체 객체 인식 모델 구현 및 평가
  - (Optional)경량화 모델 등 개별 요구사항에 적합한 모델 탐색 및 성능 검증

DAY 2 (MINI PROJECT)

WEB-CAM 기반 객체 인식 (IF NEEDED)

- YOLOv8 기반 데이터 수집/학습/deploy (Detection Alert)
  - 감시용 데이터 수집(rc_car, dummy, 등)
  - 감시용 데이터 라벨링
  - YOLOv8 기반 학습
  - YOLOv8 Object Detection

AMR-CAM 기반 객체 인식

- AMR(Autonomous Mobile Robot) Turtlebot4 개발 환경 구축
- 로봇 개발 환경에 완성 모델 서빙 및 테스트 / 로봇 H/W, 제반 환경의 한계점 도출
  - Tracking 데이터 수집((rc_car, dummy, 등)
  - Tracking 데이터 라벨링
  - YOLOv8 기반 학습
  - YOLOv8 Object Tracking

DAY 3 (MINI PROJECT)

- Auto. Driving 시스템 학습
  - Digital Mapping of environment
  - Operate AMR (Sim. & Real)
  - Tutorial 실행
  - Detection, Depth and AMR 주행
  - 로봇 개발 환경에 적용 및 테스트 / 로봇 H/W, 제반 환경의 한계점 도출

TURTLEBOT4 시뮬레이션 DEMO

- SLAM과 AutoSLAM으로 맵 생성
- Sim. Tutorial 실행
- Detection, Depth and AMR 주행 example

DAY 3 (MINI PROJECT)

REAL ROBOT

- Manually operating the AMR (Teleops)
- autonomous driving 시스템 with obstacle avoidance
  - Digital Mapping of environment
  - Launching Localization, Nav2, and using Rviz to operate a robot
  - Goal Setting and Obstacle Avoidance using Navigation

TUTORIAL

- Turtlebot4 API를 활용한 Initial Pose Navigate_to Pose 구현
- Turtlebot4 API를 활용한 Navigate_Through_pose, Follow Waypoints 구현

DAY 4 (MINI PROJECT)

- System(High Level) Design (Mini Project)
  - System Architectural Diagram
- Detail Design to Acceptance - Agile Development (SPRINTs)
  - Detection
  - AMR Control

DAY 4 (MINI PROJECT)

CODING, TEST & INTEGRATION

- Coding and Test all modules
- Porting to ROS
- And finally, Integration and Test of Detection Alert & AMR Controller

MINI PROJECT DEMO

- Prepare and demo completed project

프로젝트 RULE NUMBER ONE!!!

Have Fun Fun Fun!

HOMEWORK CHECK

- Object Detection
  - Collect various datasets (i.e. different topics/images sizes)
  - Create various models (i.e. v5, v8, v11, etc; arg: Epoch, Batch, Imgsz, augmentation, etc)
  - Analyze the results
  - Determine using key metrics which model best fit your solution
  - Using .pt file to predict/inference on pc
- Depth
  - Find and display the distance to the center of the detected objects
- Update System Requirement

SYSTEM REQUIREMENT PRESENTATION BY EACH TEAM

Using the posted notes and flipchart as needed

KEY SUBSYSTEM (MODULES) TO DEVELOP

- Detection Alert
  - Camera Capture
  - Object Detection
  - Send messages to other subsystems

- AMR Controller
  - Receive messages and act accordingly
  - Move using (SLAM) with Obstruction avoidance
  - Target Acquisition (Obj. Det.) and Tracking
  - Approach target using camera and motor control

OPERATING AMR

AMR (TURTLEBOT4)

- Features · User Manual
- https://turtlebot.github.io/turtlebot4-user-manual/overview/features.html
- Review the content

SIMULATION DEMO

TUTORIAL(SIM)

- TurtleBot 4 Navigator · User Manual

  https://turtlebot.github.io/turtlebot4-user-manual/tutorials/turtlebot4_navigator.html

SETUP BASH

- Make sure bashrc has:
  - ROS_DOMAIN_ID = 0
- Make sure discovery setup.bash is **not** sourced!
- source ~/.bashrc

OPERATING A ROBOT(SIM) – GAZEBO

TERM1

- ros2 launch turtlebot4_ignition_bringup turtlebot4_ignition.launch.py

TERM2

- ros2 topic list
- ros2 topic echo <topic> --once
  - /oakd/rgb/preview/image_raw
  - /oakd/rgb/preview/depth
  - ….

OPERATING A ROBOT(SIM)

- Dock/Undock
- Manual Driving
  - Teleops
- Camera Display
  - RGB/Depth

- Navigation with rviz
  - 2D_Pose_Estimate (initial position)
  - Nav2_Goal

DIGITAL MAPPING USING SLAM (SIM)

TERM1

- ros2 launch turtlebot4_ignition_bringup turtlebot4_ignition.launch.py **nav2:=true slam:=true rviz:=true**

ON GAZEBO

- Undock the robot
- Use keyboard to operate and complete the map

DIGITAL MAPPING WITH AUTO – SLAM (SIM)

TERM1

-
  ```
  ros2 launch turtlebot4_ignition_bringup turtlebot4_ignition.launch.py nav2:=true slam:=true rviz:=true
  ```
- Undock the robot
- Set init pose from rviz

TERM2

-
  ```
  ros2 launch explore_lite explore.launch.py
  ```

TUTORIAL(SIM)

TERMINAL 1

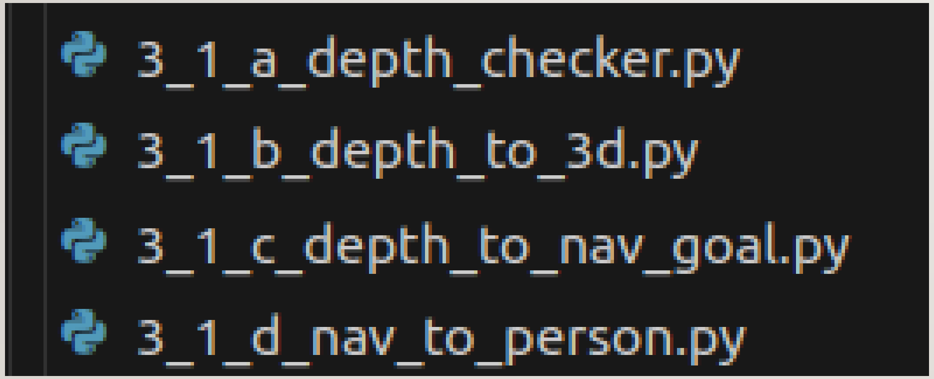
```
$ ros2 launch turtlebot4_ignition_bringup turtlebot4_ignition.launch.py nav2:=true slam:=false localization:=true rviz:=true
```

- Undock and set init pose

TERMINAL 2

```
$ ros2 run turtlebot4_python_tutorials nav_to_pose
$ ros2 run turtlebot4_python_tutorials nav_through_poses
$ ros2 run turtlebot4_python_tutorials follow_waypoints
$ ros2 run turtlebot4_python_tutorials create_path
$ ros2 run turtlebot4_python_tutorials mail_delivery
$ ros2 run turtlebot4_python_tutorials patrol_loop
```

USING DEPTH (SIM)



3_1_a_depth_checker.py

3_1_b_depth_to_3d.py

3_1_c_depth_to_nav_goal.py

3_1_d_nav_to_person.py

TERMINAL 2 (

```
$ ros2 run <pkg_name> <exec_name>
```

For example,

```
$ ros2 run day3 nav_to_person
```

OPERATING THE ROBOT (AUTONOMOUSLY)

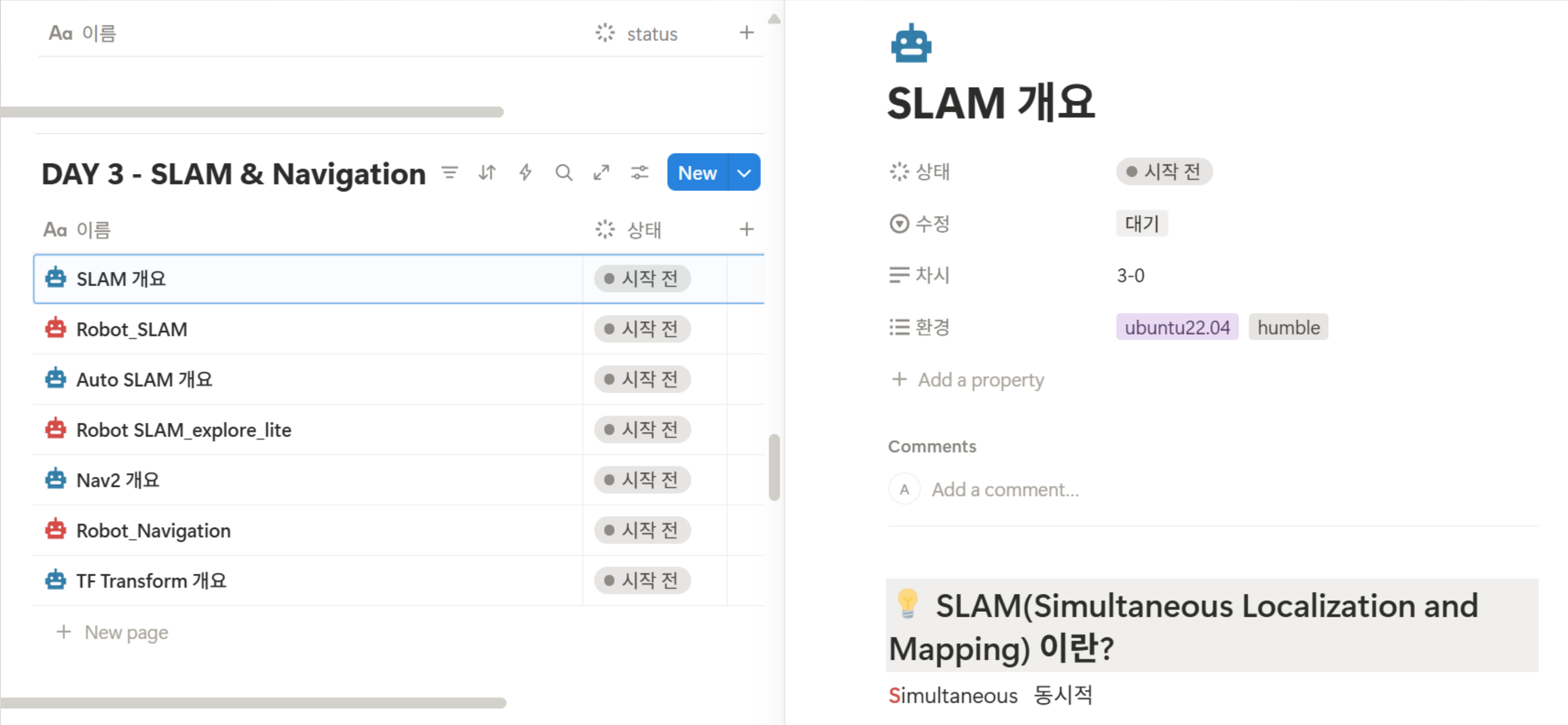
SETUP BASH

- Make sure bashrc has:
  - ROS_DOMAIN_ID = 0
- echo “alias **ros-restart**=‘ros2 daemon stop; ros2 daemon start’" >> ~/.bashrc
- Make sure discovery setup.bash **is** sourced!
- source ~/.bashrc

DIGITAL MAPPING (SLAM)

- Generating a map · User Manual https://turtlebot.github.io/turtlebot4-user-manual/tutorials/generate_map.html

SLAM 개요



Aa 이름 status +

DAY 3 - SLAM & Navigation

Aa 이름	상태
SLAM 개요	● 시작 전
Robot_SLAM	● 시작 전
Auto SLAM 개요	● 시작 전
Robot SLAM_explore_lite	● 시작 전
Nav2 개요	● 시작 전
Robot_Navigation	● 시작 전
TF Transform 개요	● 시작 전

+ New page

SLAM 개요

상태	● 시작 전
수정	대기
차시	3-0
환경	ubuntu22.04 humble

+ Add a property

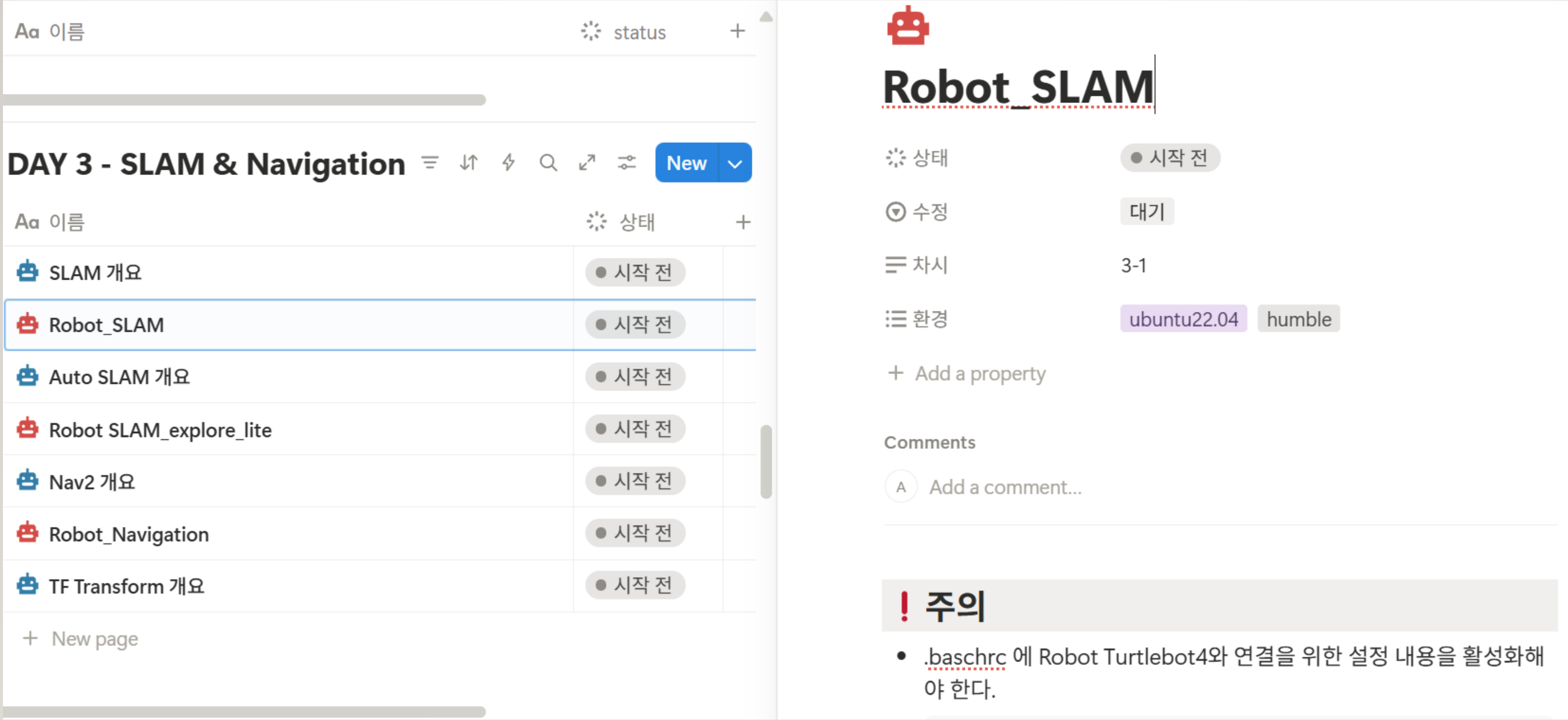
Comments

Add a comment...

💡 SLAM(Simultaneous Localization and Mapping) 이란?

Simultaneous 동시적

SLAM WITH ROBOT



Aa 이름	status

DAY 3 - SLAM & Navigation

Aa 이름	상태
SLAM 개요	● 시작 전
Robot_SLAM	● 시작 전
Auto SLAM 개요	● 시작 전
Robot SLAM_explore_lite	● 시작 전
Nav2 개요	● 시작 전
Robot_Navigation	● 시작 전
TF Transform 개요	● 시작 전

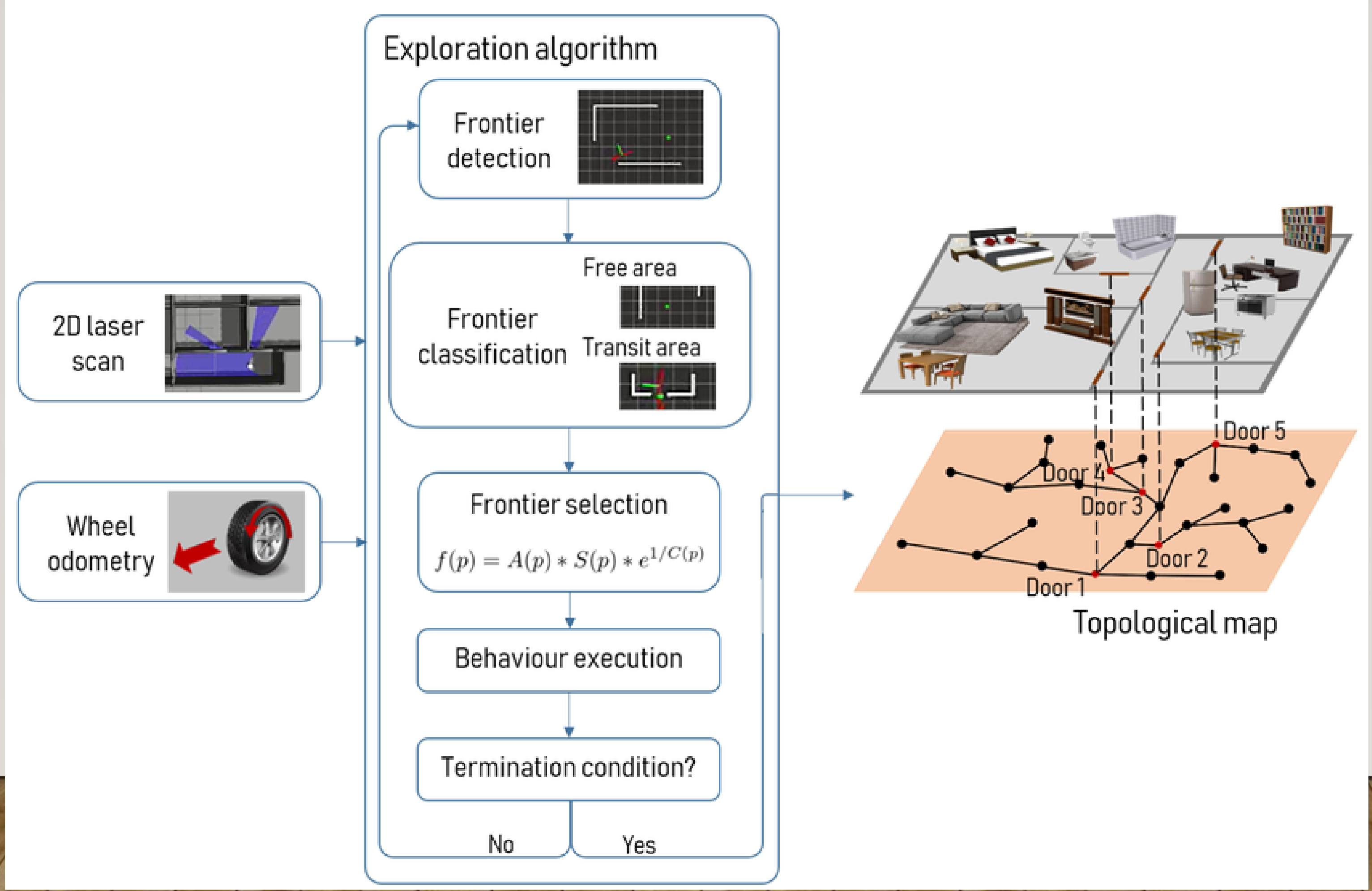
Robot_SLAM

상태	● 시작 전
수정	대기
차시	3-1
환경	ubuntu22.04 humble

❗ 주의

- .baschrc 에 Robot Turtlebot4와 연결을 위한 설정 내용을 활성화해야 한다.

AUTO SLAM CONCEPT/ALGORITHM



Exploration algorithm

2D laser scan

Wheel odometry

Frontier detection

Frontier classification

Free area

Transit area

Frontier selection

$f(p) = A(p) * S(p) * e^{1/C(p)}$

Behaviour execution

Termination condition?

No

Yes

ALGORITHM DETAIL

- Map Subscription

explore_lite subscribes to the SLAM-generated occupancy grid (/map topic) and identifies:

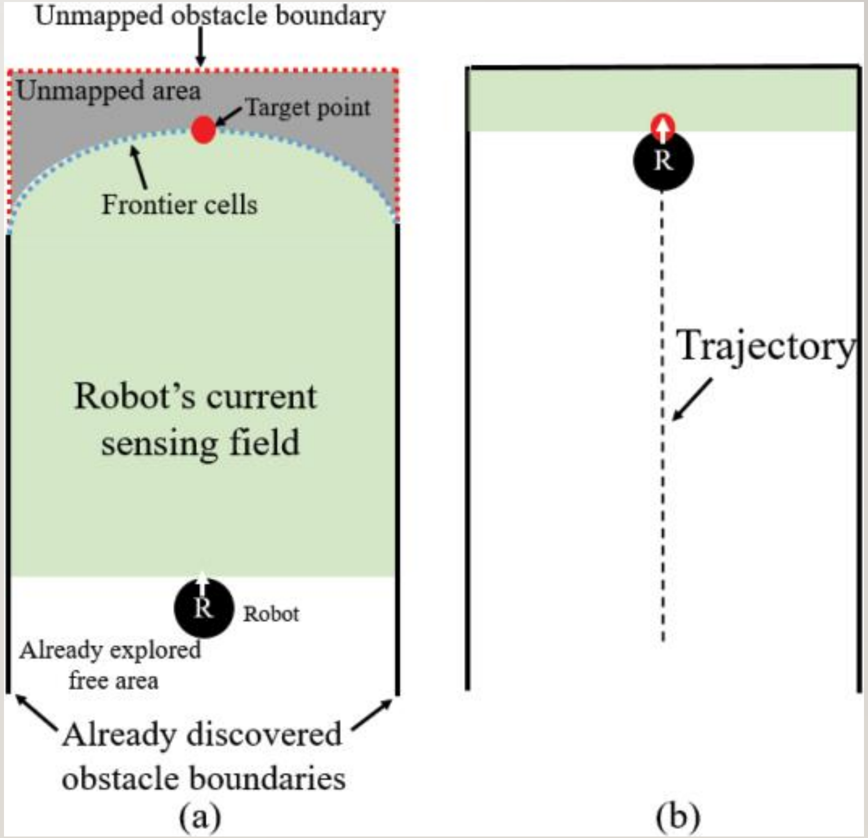
  - Free space: known, unoccupied areas
  - Occupied space: obstacles
  - Unknown space: unexplored

- Frontier Detection

The map is scanned for cells that:

  - Are free, and
  - Are adjacent to at least one unknown cell.

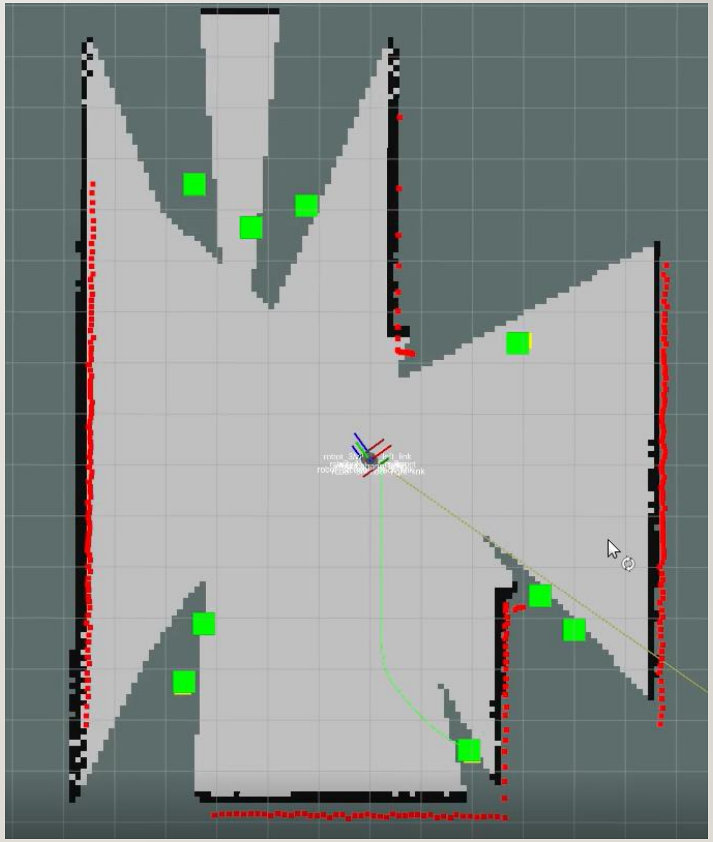
These are marked as frontier cells.


Unmapped obstacle boundary
Unmapped area
Target point
Frontier cells
Robot's current sensing field
R
Robot
Already explored free area
Already discovered obstacle boundaries
(a)
R
Trajectory
(b)

ALGORITHM DETAIL

- Frontier Grouping
  - Frontier cells are clustered into connected regions.
  - Each group represents a potential exploration target.

- Goal Selection
  - For each frontier group, a representative point (typically the centroid or closest point) is selected.
  - The robot scores each group based on:
    - Distance from the robot
    - Information gain (how much new area might be revealed)
  - The best-scoring frontier is chosen as the next goal.

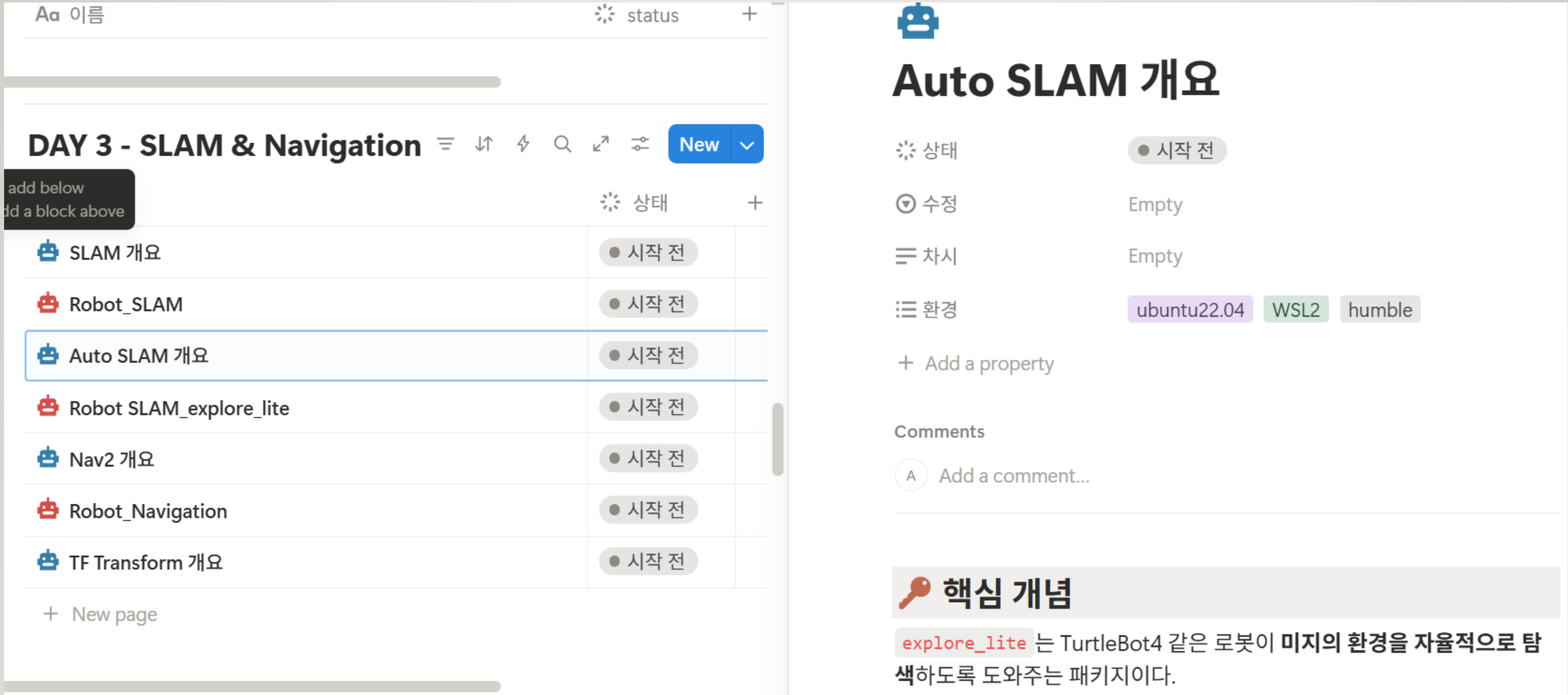


ALGORITHM DETAIL

- Termination

```
While (frontiers exist and reachable)
    Select best frontier
    Send as goal
    If goal fails → blacklist
If (no frontiers or all blacklisted)
    Terminate exploration
```

AUTO SLAM 개요



DAY 3 - SLAM & Navigation

	상태
SLAM 개요	● 시작 전
Robot_SLAM	● 시작 전
Auto SLAM 개요	● 시작 전
Robot SLAM_explore_lite	● 시작 전
Nav2 개요	● 시작 전
Robot_Navigation	● 시작 전
TF Transform 개요	● 시작 전

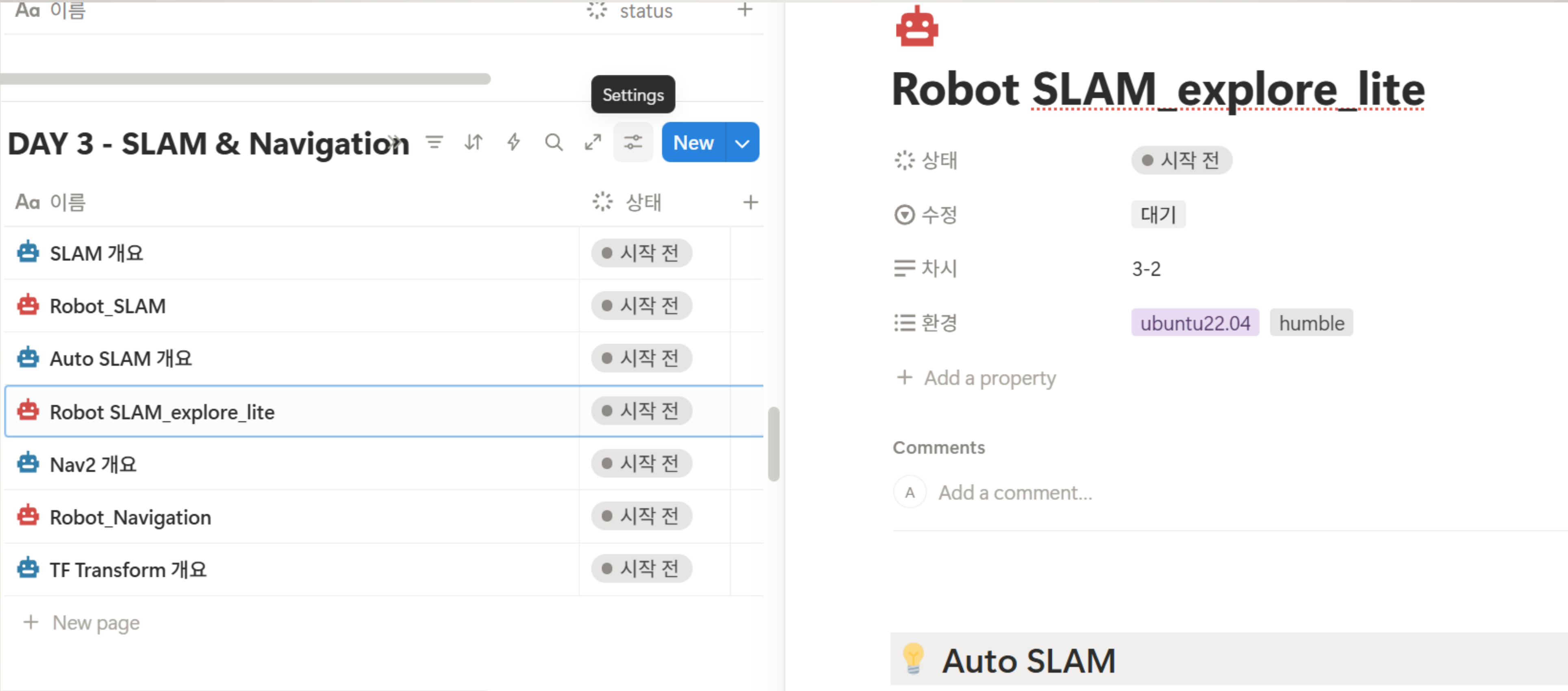
Auto SLAM 개요

상태	● 시작 전
수정	Empty
차시	Empty
환경	ubuntu22.04 WSL2 humble

🔑 핵심 개념

`explore_lite`는 TurtleBot4 같은 로봇이 **미지의 환경을 자율적으로 탐색**하도록 도와주는 패키지이다.

AUTO SLAM



DAY 3 - SLAM & Navigation

Aa 이름	상태
SLAM 개요	● 시작 전
Robot_SLAM	● 시작 전
Auto SLAM 개요	● 시작 전
Robot SLAM_explore_lite	● 시작 전
Nav2 개요	● 시작 전
Robot_Navigation	● 시작 전
TF Transform 개요	● 시작 전

Robot SLAM_explore_lite

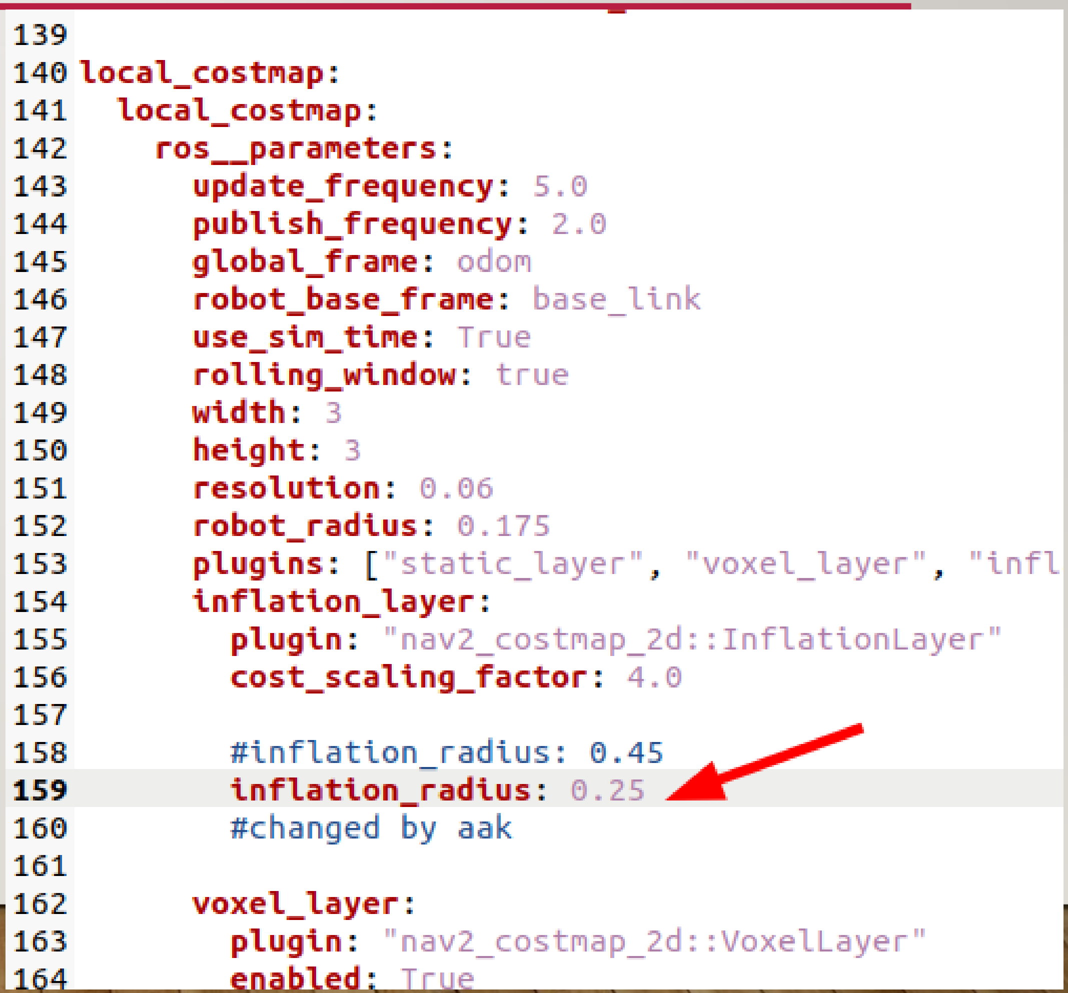
상태	● 시작 전
수정	대기
차시	3-2
환경	ubuntu22.04 humble

💡 Auto SLAM

SIMPLE NAV2 PARAM ADJUSTMENT

$ cd ~/turtlebot4_ws/src/turtlebot4/turtlebot4_navigation/config

- Change/adjust "inflation_radius" to fit your environment



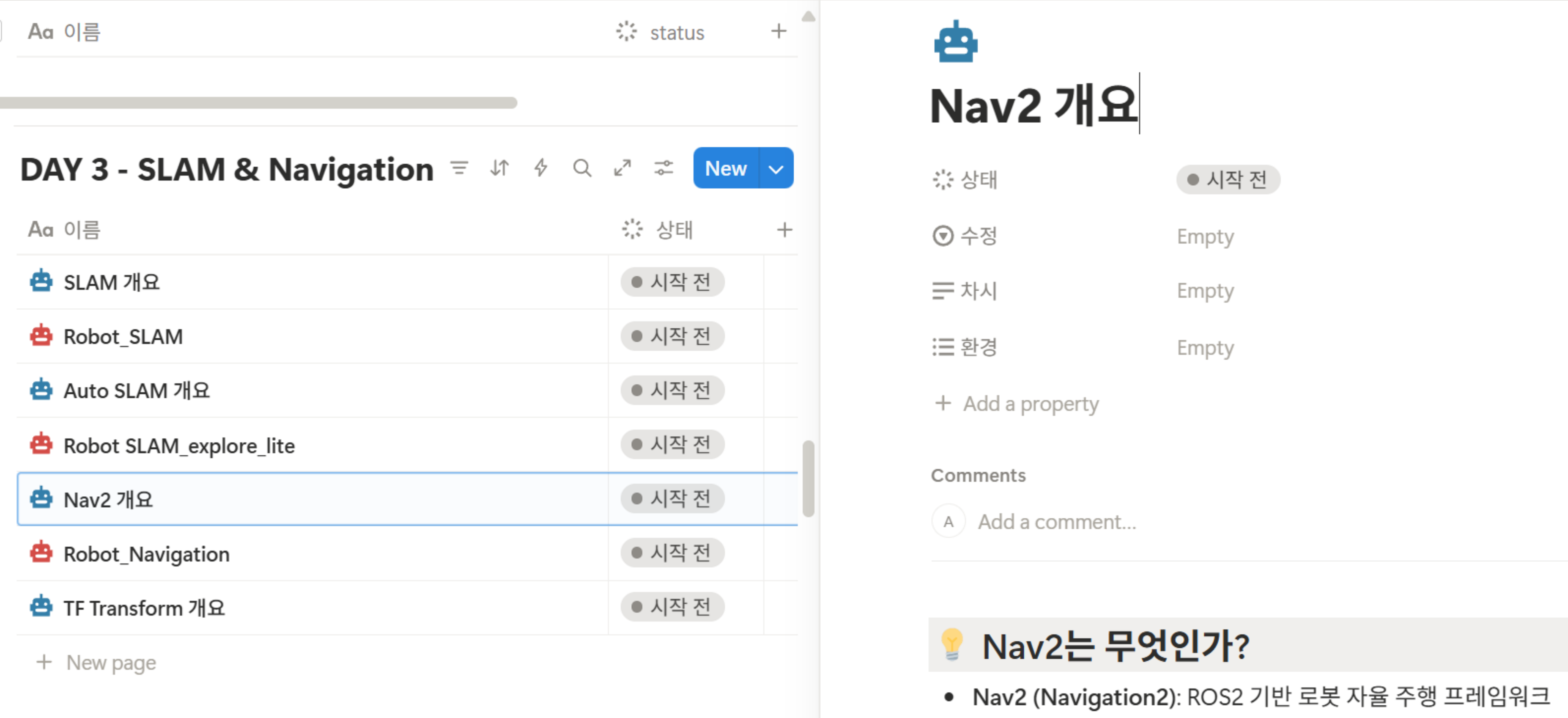
```

local_costmap:
  local_costmap:
    ros__parameters:
      update_frequency: 5.0
      publish_frequency: 2.0
      global_frame: odom
      robot_base_frame: base_link
      use_sim_time: True
      rolling_window: true
      width: 3
      height: 3
      resolution: 0.06
      robot_radius: 0.175
      plugins: ["static_layer", "voxel_layer", "infl
      inflation_layer:
        plugin: "nav2_costmap_2d::InflationLayer"
        cost_scaling_factor: 4.0

        #inflation_radius: 0.45
        inflation_radius: 0.25
        #changed by aak

      voxel_layer:
        plugin: "nav2_costmap_2d::VoxelLayer"
        enabled: True
```

NAV2 개요



Aa 이름	status

DAY 3 - SLAM & Navigation

Aa 이름	상태
SLAM 개요	● 시작 전
Robot_SLAM	● 시작 전
Auto SLAM 개요	● 시작 전
Robot SLAM_explore_lite	● 시작 전
Nav2 개요	● 시작 전
Robot_Navigation	● 시작 전
TF Transform 개요	● 시작 전

Nav2 개요

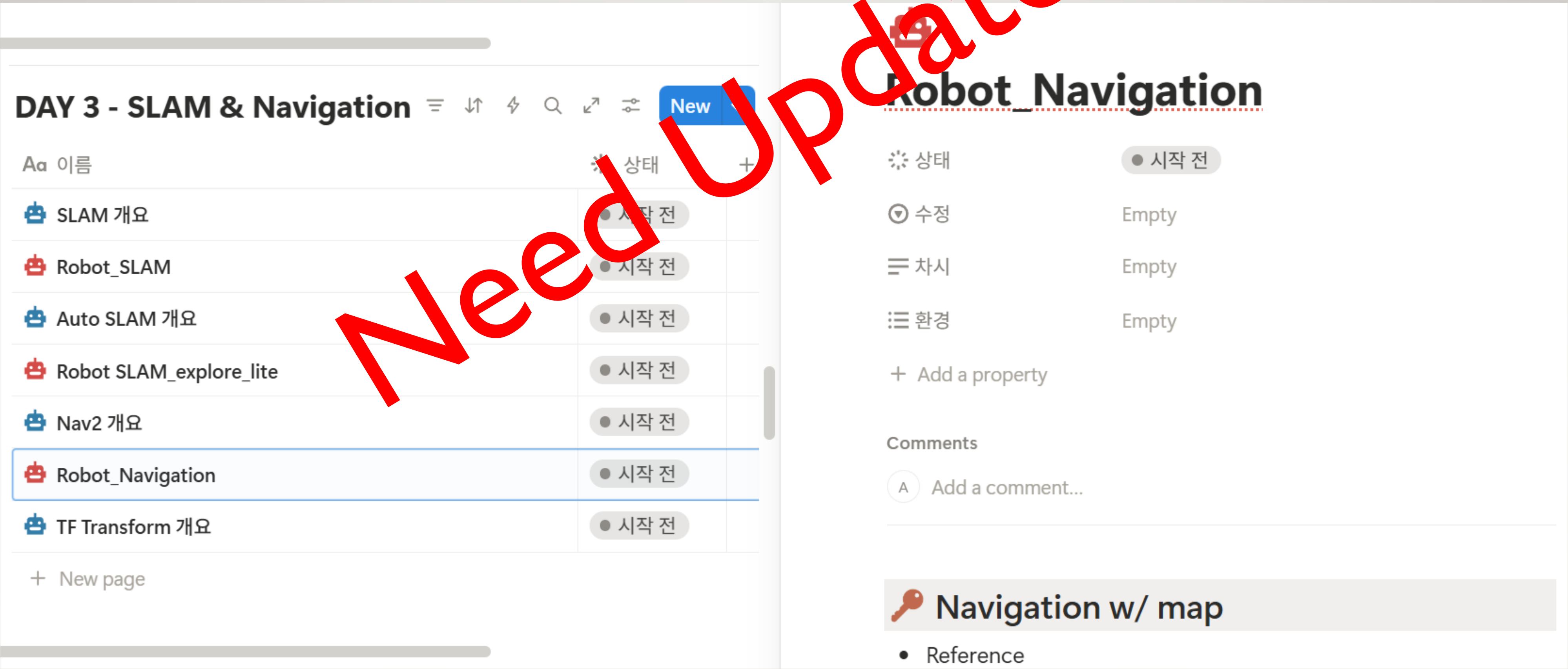
상태	● 시작 전
수정	Empty
차시	Empty
환경	Empty

Nav2는 무엇인가?

- Nav2 (Navigation2): ROS2 기반 로봇 자율 주행 프레임워크

NAV2

Need Update



DAY 3 - SLAM & Navigation

New

Aa 이름	상태
SLAM 개요	● 시작 전
Robot_SLAM	● 시작 전
Auto SLAM 개요	● 시작 전
Robot SLAM_explore_lite	● 시작 전
Nav2 개요	● 시작 전
Robot_Navigation	● 시작 전
TF Transform 개요	● 시작 전

+ New page

Robot_Navigation

상태	● 시작 전
수정	Empty
차시	Empty
환경	Empty

+ Add a property

Comments

Add a comment...

Navigation w/ map

- Reference

TUTORIAL EXERCISE

Make copy and Update the tutorial code to successfully execute in the project environment with real robot

Tutorial Codes are found in:

$HOME/turtlebot4_ws/src/turtlebot4_tutorials/turtlebot4_python_tutorials/turtlebot4_python_tutorials

TUTORIAL

TERMINAL 1

```
$ ros2 launch turtlebot4_navigation localization.launch.py namespace:=/robot<n> map:=$HOME/Documents/room/room_map.yaml
```

TERMINAL 2

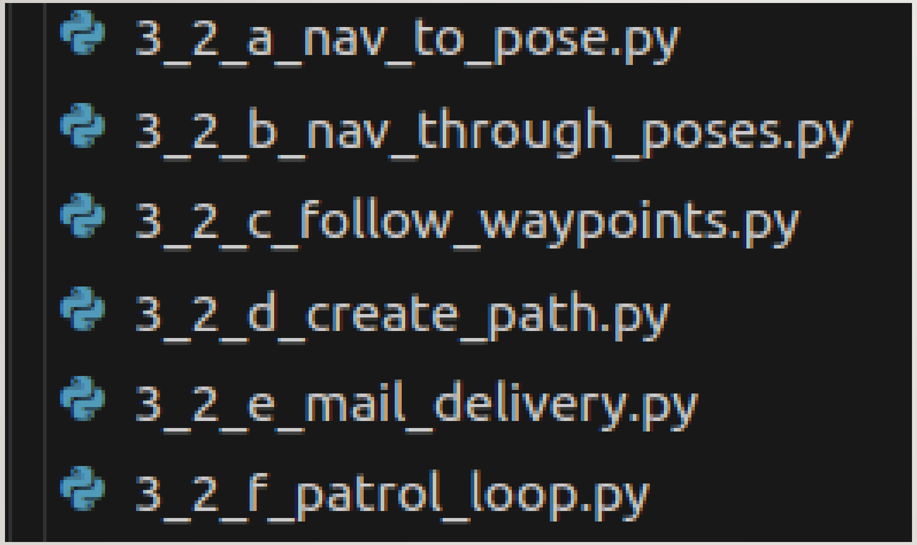
```
$ ros2 launch turtlebot4_viz view_robot.launch.py namespace:=/robot <n>
```

- Undock and set init pose

TERMINAL 3

```
$ ros2 launch turtlebot4_navigation nav2.launch.py namespace:=/robot <n>
```

TUTORIAL



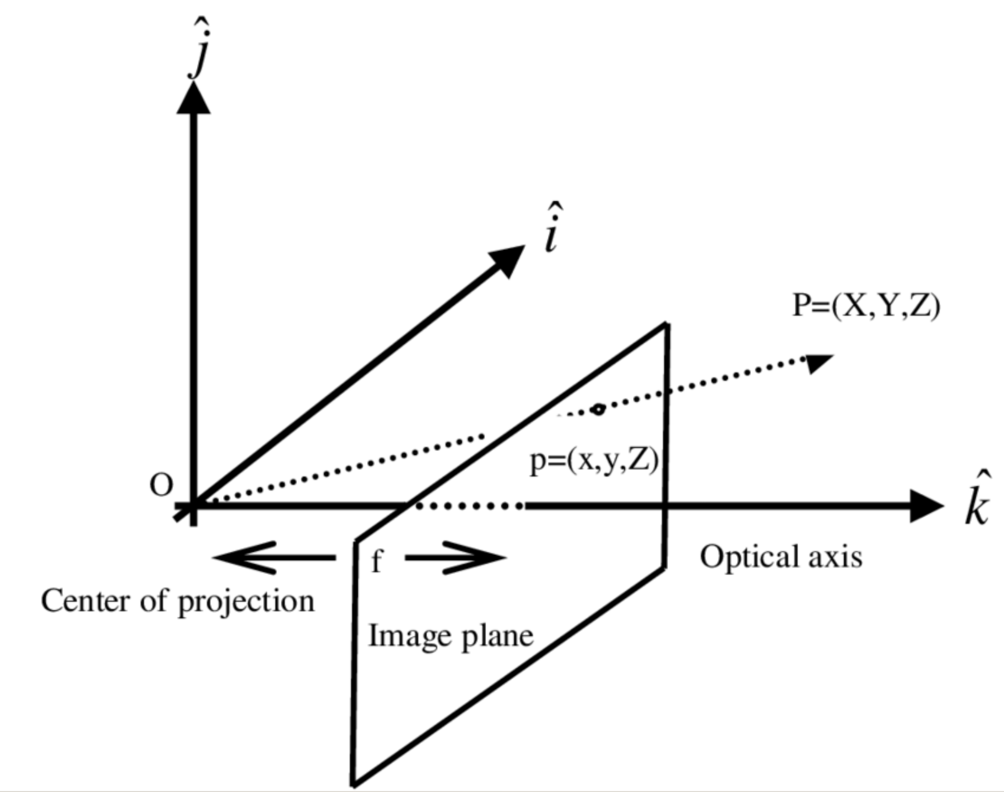
3_2_a_nav_to_pose.py
3_2_b_nav_through_poses.py
3_2_c_follow_waypoints.py
3_2_d_create_path.py
3_2_e_mail_delivery.py
3_2_f_patrol_loop.py

TERMINAL 4

```
$ ros2 run day3 create_path --ros-args -r __ns:=/robot<n>
$ ros2 run day3 nav_to_poses --ros-args -r __ns:=/robot<n>
$ ros2 run day3 follow_waypoints --ros-args -r __ns:=/robot<n>
$ ros2 run day3 nav_through_poses --ros-args -r __ns:=/robot<n>
$ ros2 run day3 mail_delivery --ros-args -r __ns:=/robot<n>
$ ros2 run day3 patrol_loop --ros-args -r __ns:=/robot<n>
```

USING DEPTH TO GET COORDINATES (TF TRANSFORM)

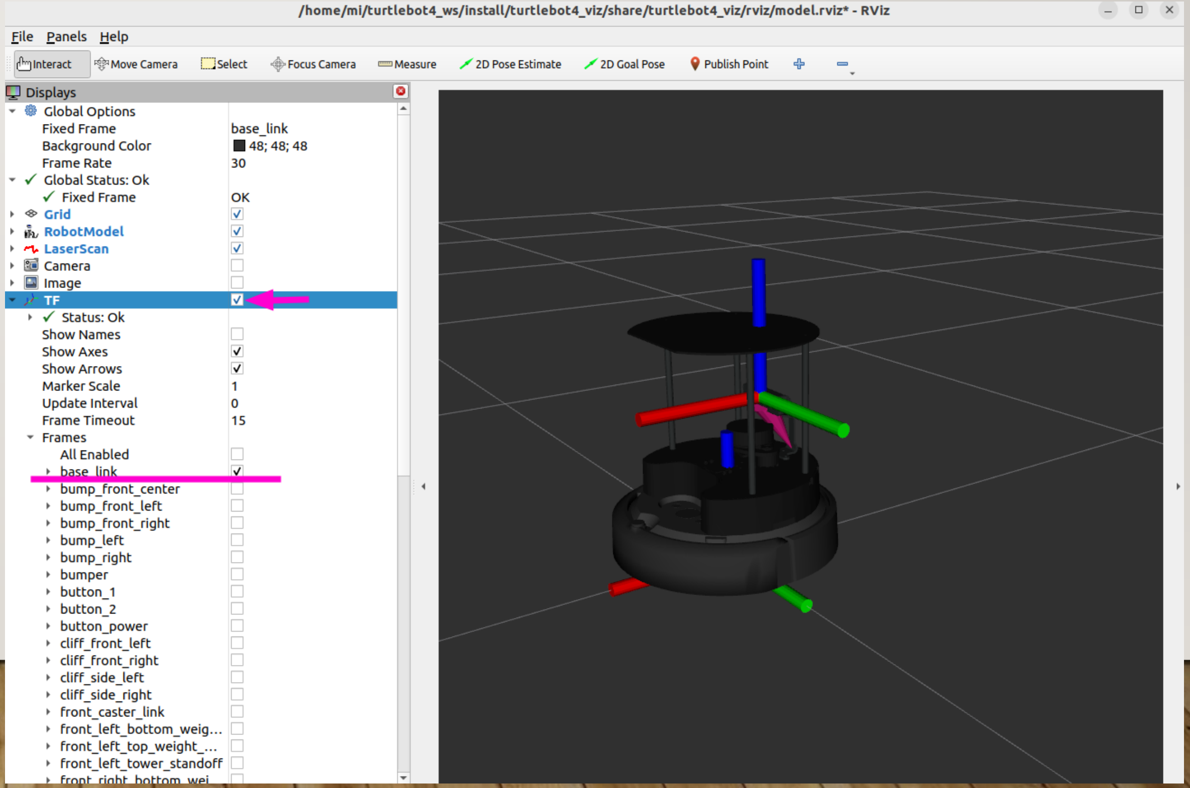
CAMERA INTRINSIC AND REPROJECTION



$$X = \frac{(u - c_x) \cdot Z}{f_x}, \quad Y = \frac{(v - c_y) \cdot Z}{f_y}, \quad Z = Z$$

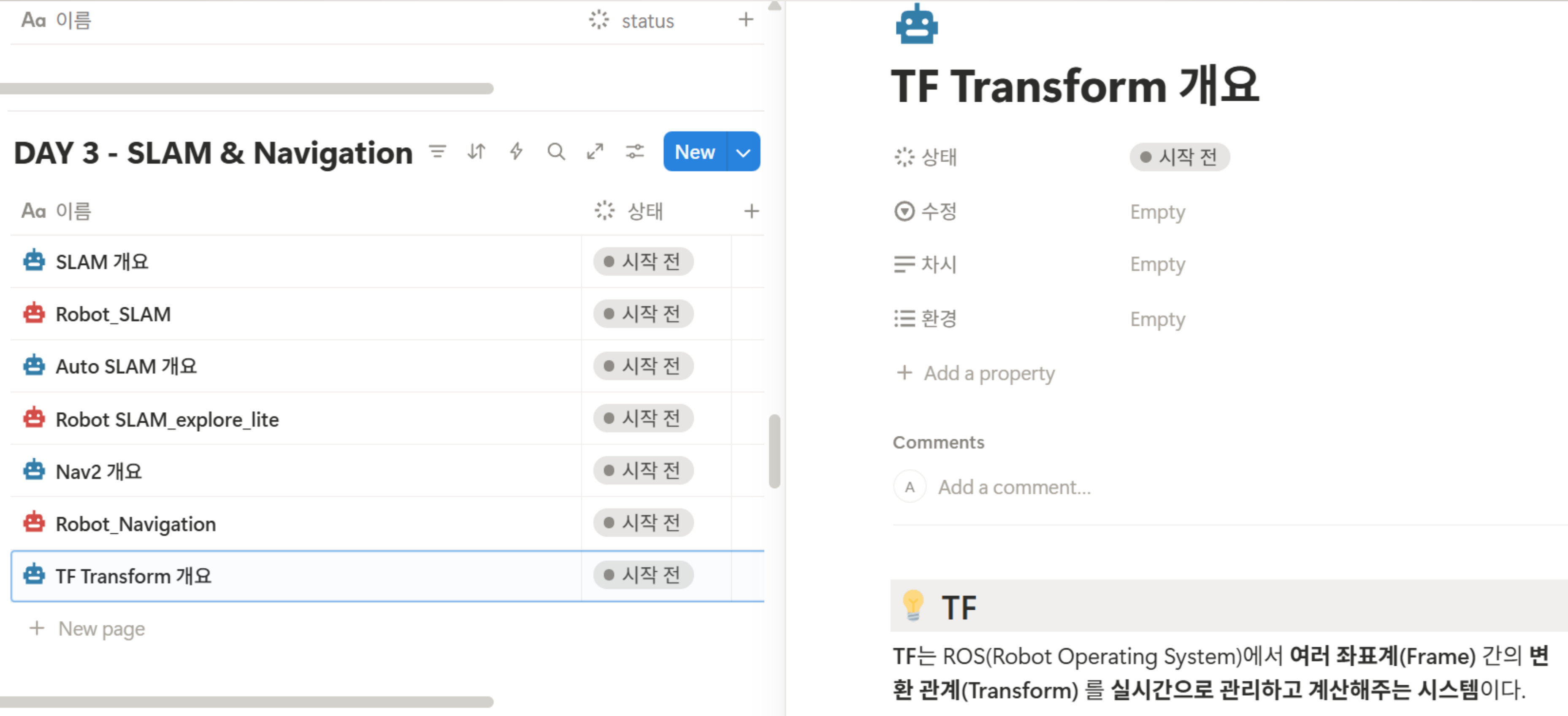
ROS TRANSFORM EXPLAINED

- Transform Explained (1)
- https://indecisive-freedom-6e8.notion.site/Transform-Explained-1-22f8e215779c809eb2d4cb0be5538056



Global Options	
Fixed Frame	base_link
Background Color	■ 48; 48; 48
Frame Rate	30
✓ Global Status: Ok	
✓ Fixed Frame	OK
Grid	☑
RobotModel	☑
LaserScan	☑
Camera	□
Image	□
TF	☑
✓ Status: Ok	
Show Names	□
Show Axes	☑
Show Arrows	☑
Marker Scale	1
Update Interval	0
Frame Timeout	15
Frames	
All Enabled	□
base_link	☑
bump_front_center	□
bump_front_left	□
bump_front_right	□
bump_left	□
bump_right	□
bumper	□
button_1	□
button_2	□
button_power	□
cliff_front_left	□
cliff_front_right	□
cliff_side_left	□
cliff_side_right	□
front_caster_link	□
front_left_bottom_weig…	□
front_left_top_weight_…	□
front_left_tower_standoff	□
front_right_bottom_wei	□

TF TRANSFORM 개요



Aa 이름	status

DAY 3 - SLAM & Navigation

Aa 이름	상태
SLAM 개요	● 시작 전
Robot_SLAM	● 시작 전
Auto SLAM 개요	● 시작 전
Robot SLAM_explore_lite	● 시작 전
Nav2 개요	● 시작 전
Robot_Navigation	● 시작 전
TF Transform 개요	● 시작 전

TF Transform 개요

상태	● 시작 전
수정	Empty
차시	Empty
환경	Empty

TF

TF는 ROS(Robot Operating System)에서 **여러 좌표계(Frame)** 간의 **변환 관계(Transform)** 를 **실시간으로 관리하고 계산해주는 시스템**이다.

SETUP BASH(ROBOT)

- Make sure bashrc has:
  - ROS_DOMAIN_ID = 0

- Make sure discovery setup.bash **is** sourced!

- source ~/.bashrc

USING DEPTH (ROBOT)

TERMINAL 1

```
$ ros2 launch turtlebot4_navigation localization.launch.py namespace:=/robot<n> map:=$HOME/Documents/room/room_map.yaml
```

TERMINAL 2

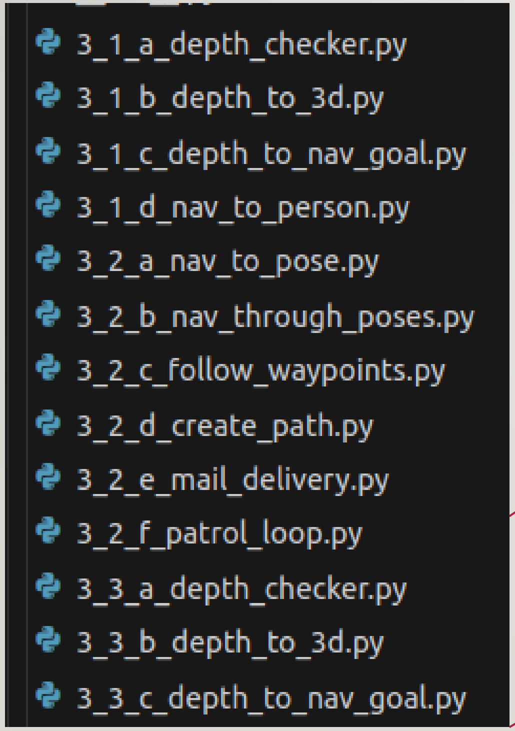
```
$ ros2 launch turtlebot4_viz view_robot.launch.py namespace:=/robot <n>
```

- Set Init Pose using 2D_PoseEstimate

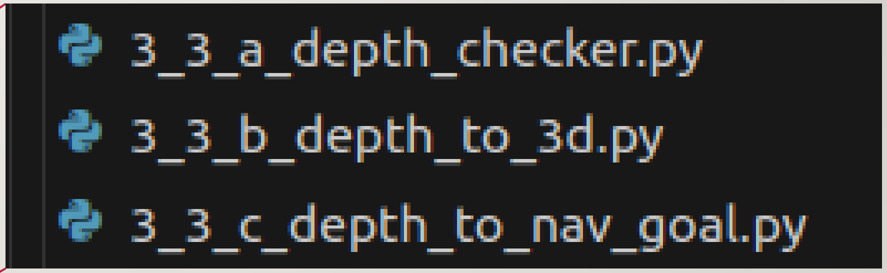
TERMINAL 3

```
$ ros2 launch turtlebot4_navigation nav2.launch.py namespace:=/robot <n>
```

USING DEPTH (ROBOT)



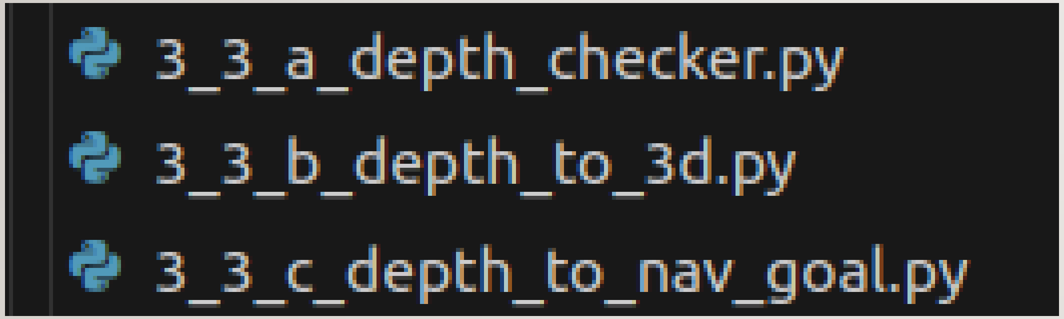
```
3_1_a_depth_checker.py
3_1_b_depth_to_3d.py
3_1_c_depth_to_nav_goal.py
3_1_d_nav_to_person.py
3_2_a_nav_to_pose.py
3_2_b_nav_through_poses.py
3_2_c_follow_waypoints.py
3_2_d_create_path.py
3_2_e_mail_delivery.py
3_2_f_patrol_loop.py
3_3_a_depth_checker.py
3_3_b_depth_to_3d.py
3_3_c_depth_to_nav_goal.py
```



```
3_3_a_depth_checker.py
3_3_b_depth_to_3d.py
3_3_c_depth_to_nav_goal.py
```

Exercise: using the simulation code develop the code for the actual robot

USING DEPTH (ROBOT)



3_3_a_depth_checker.py

3_3_b_depth_to_3d.py

3_3_c_depth_to_nav_goal.py

```
$ ros2 run day4 depth_checker --ros-args -r __ns:=/robot<n>
```

```
$ ros2 run day4 depth_to_3d --ros-args -r __ns:=/robot<n> -r /tf:=/robot<n>/tf –r /tf_static:=/robot<n>/tf_static
```

```
$ ros2 run day4 depth_to_goal --ros-args -r __ns:=/robot<n> -r /tf:=/robot<n>/tf –r /tf_static:=/robot<n>/tf_static
```

프로젝트 RULE NUMBER ONE!!!

Are we still having FUN!