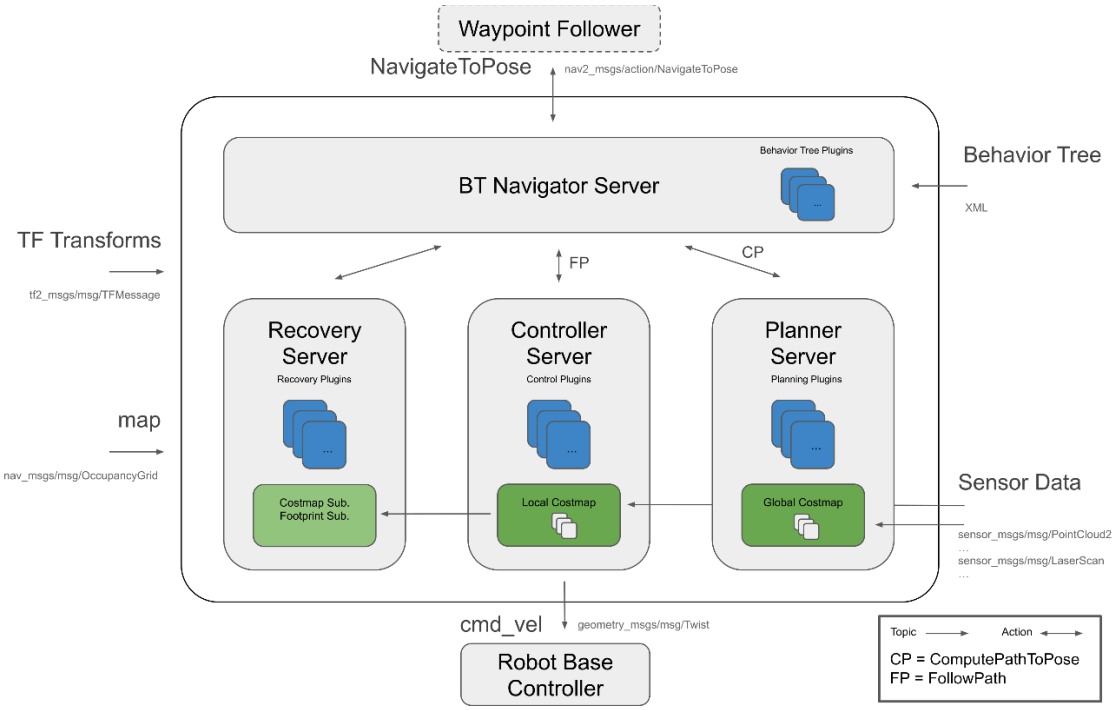
第一讲： ISAAC SIM 定制化机器人导航案例

0. 预先知识

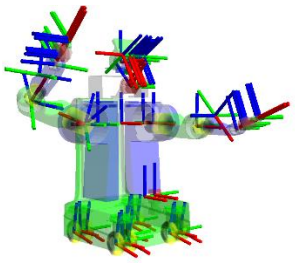
1. **ISAAC SIM <-> ROS Bridges <-> ROS ecosystem**
2. **Navigation2 Stack**



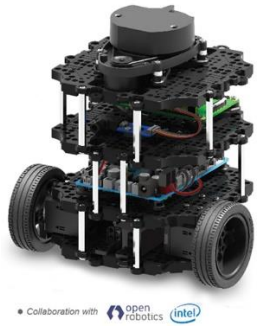
Input: (1) Map (OccupancyGrid) (2) SensorData (LaserScan) (3) TF Transforms (TFMessage)

Nav2 requires the following TF transformations to be published in ROS2:

1. map => odom (dynamic)
2. odom => base_link (dynamic)
3. base_link => base_scan (static)



1. ISAAC SIM 定制化机器人模型导入 [import_urdf_000]



1. 导入 Turtlebot robot: Import URDF to USD
    a. urdf 路径：/opt/ros/foxy/share/turtlebot3_description/urdf/turtlebot3_burger.urdf
2. Isaac Assets > Environments > Simple_warehoused 导入 warehouse_with_forklifts.usd 资产作为建图导航的 3D 环境
3. 将整体保存为 USD 文件

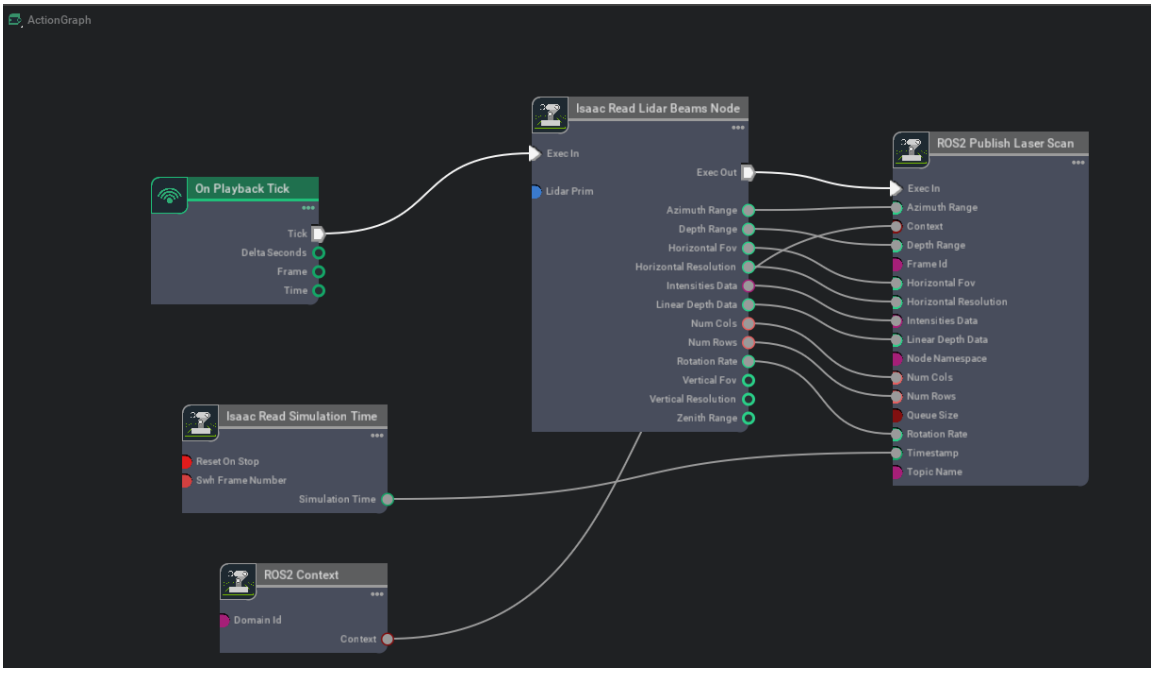
2. ISAAC SIM Occupancy Map Generatoin 2D 栅格地图自动生成 [occupancy_map_001]

1. 打开 Isaac Utils > Occupancy Map 插件
2. 设置 Center To Selection: turtlebot 的 base_scan 框架的 Z 坐标高度为 0.182 m
3. 设置 Bound Selection：选择 warehouse 的 XForm
4. CALCULATE -> VISUALIZE IMAGE -> Rotate Image: –90 -> Coordinate Type: ROS YAML -> Save as
   Isaac_sim/ros2_workspace/src/navigation/carter_navigation/maps/carter_warehouse_navigation.yaml
   Isaac_sim/ros2_workspace/src/navigation/carter_navigation/maps/carter_warehouse_navigation.png
   参考 12. Occupancy Map Generation — Omniverse Robotics documentation (nvidia.com)

3. ISAAC SIM ROS2 Bridge 定制化配置 [turtlebot_navigation_002]

Bring up sensors and drivers: (ActionGraph)

1. 在菜单 Window > Extension 中**切换到 ROS2 Bridge**, before you load any saved ROS2 ActionGraph
2. 添加 sensors: Lidar
3. 创建 ActionGraph of ROS2 Bridge: (Lidar, TF Trees, Raw TF Trees, Wheelbase...)
    a. Tick: (**On Playback Tick Node**)
    b. Timestamp: (**Isaac Read Simulation Time Node**)
    c. 配置 Domain ID: (**ROS2 Context Node**)
    d. Lidar: (**Isaac Read Lidar Beams Node + ROS2 Publish Laser Scan**)
        i. 配置 inputs:lidarPrim
        ii. 配置 frameId: base_scan, topicName: scan



e. Wheelbase: /cmd_vel

i. **ROS2 Subscribe Twist Node:** 配置 topicName: cmd_vel

ii. **Differential Controller Node:** 配置 wheelDistance, wheelRadius

1. WheelRadius: 0.033 m
2. WheelDistance: 0.160 m

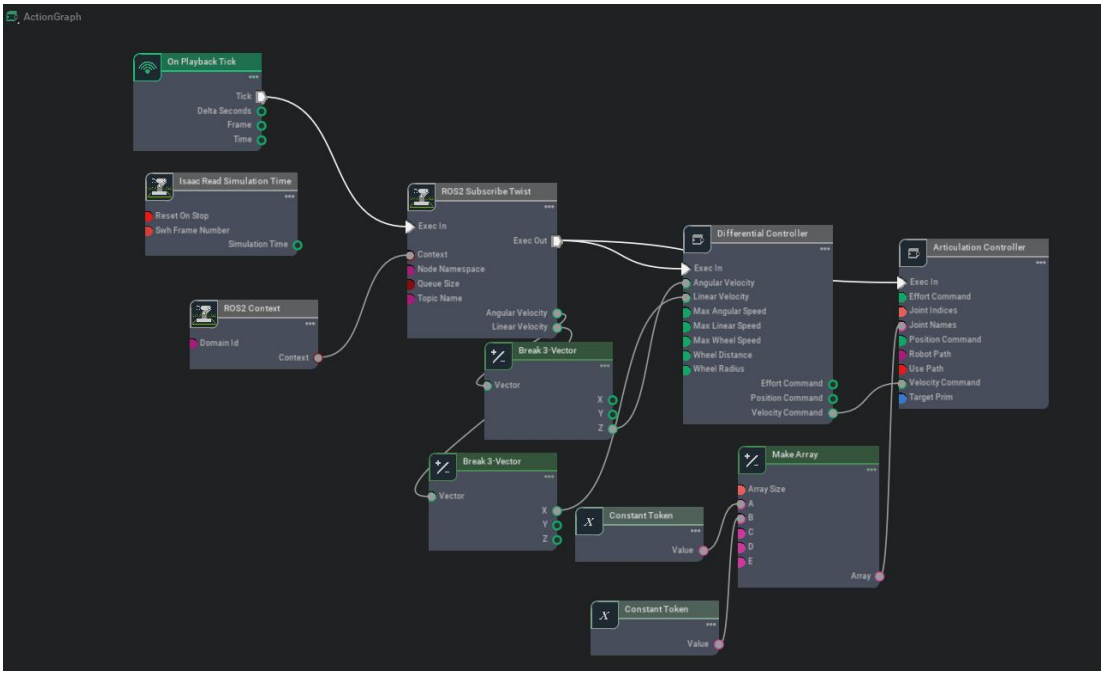
iii. **Articulation Controller Node:** 配置 inputs:TargetPrim (Articulation Root Prim), Joint Names

1. Untoggle: usePath

iv. **Break 3-Vector Node, Constant Token Node, Make Array Node**

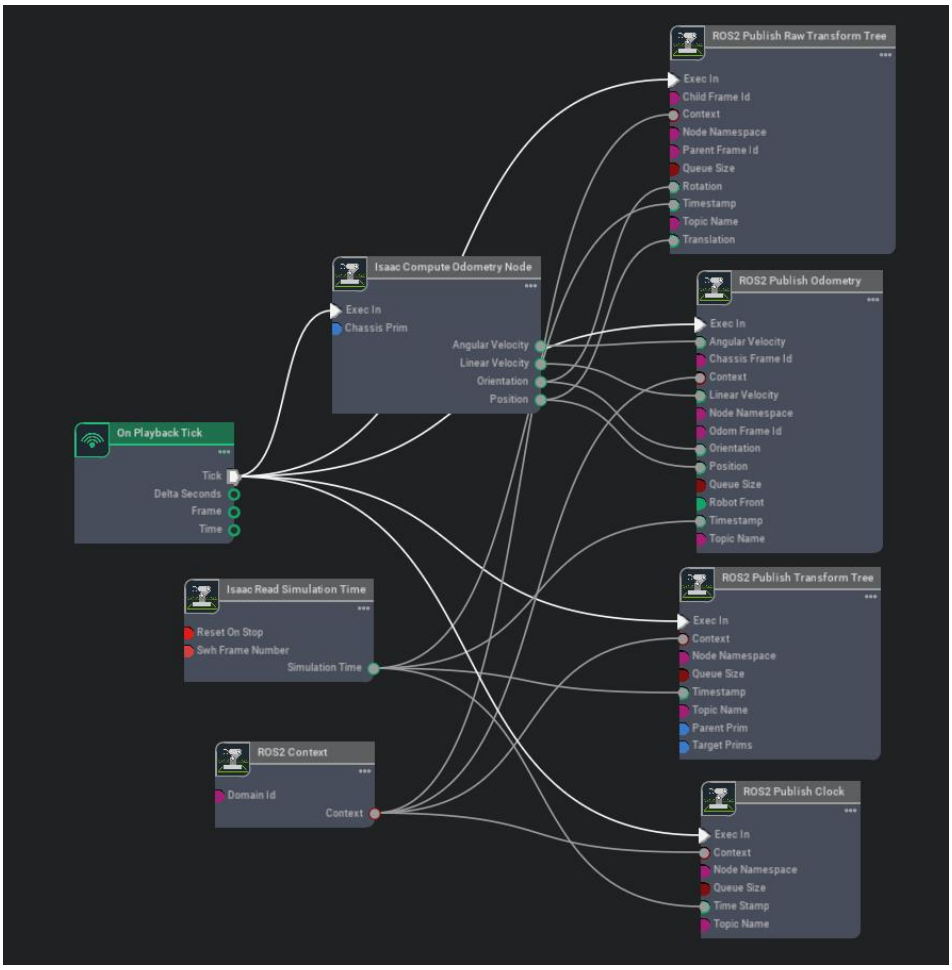
v. 测试底盘驱动是否配置成功:

ros2 topic pub /cmd_vel geometry_msgs/msg/Twist "{linear: {x: 2.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 1.8}}"

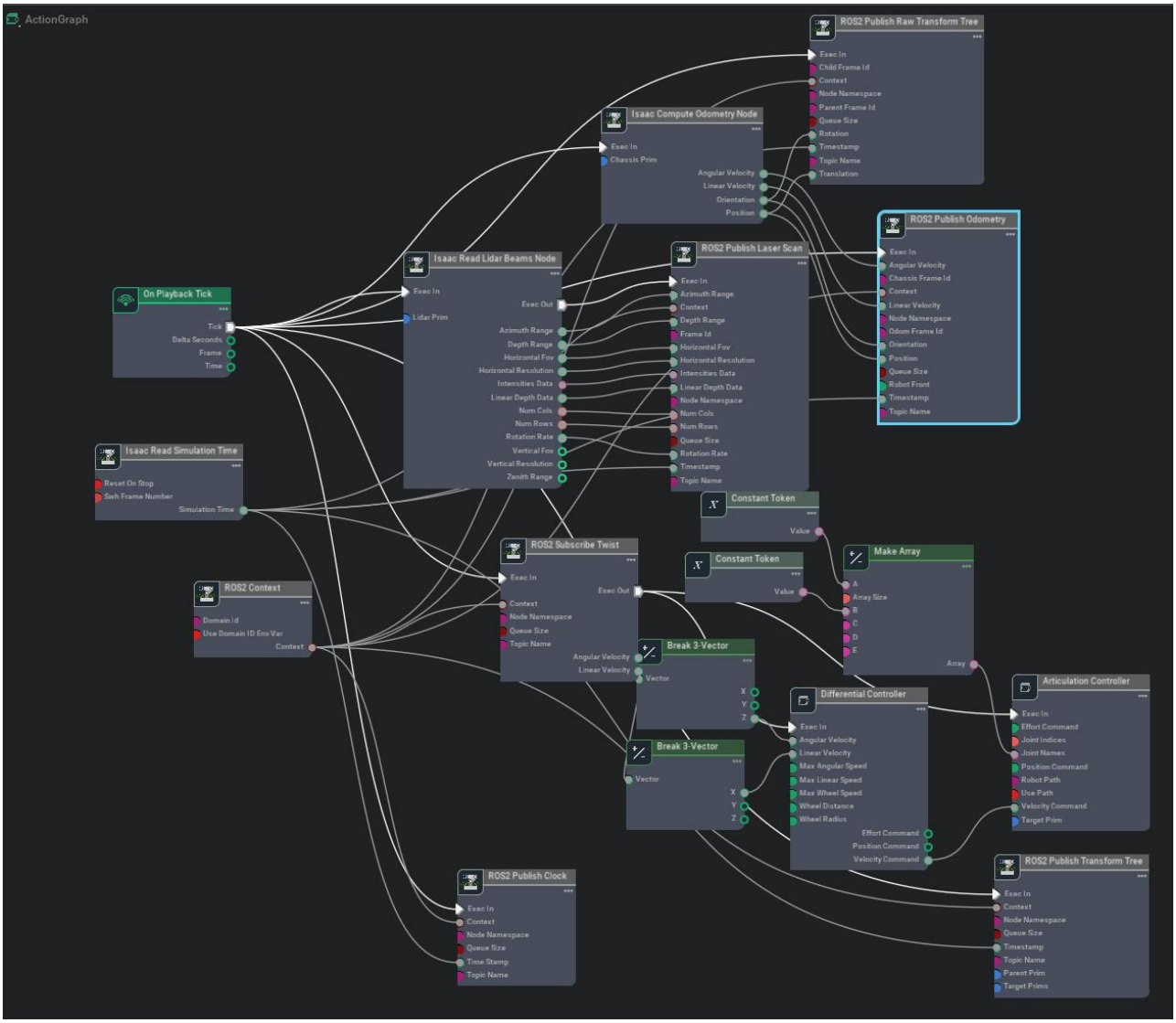


f. Odom: /odom + Raw TF Tree: /tf
   i. Raw TF: (**Isaac Compute Odometry Node + ROS2 Publish Raw Transform Tree Node**)
      1. **Isaac Compute Odometry Node:** 配置 inputs:chassisPrim (articulation root prim)
      2. **ROS2 Publish Raw Transform Tree Node**: 配置 childFrameId: base_link, parentFrameId: odom, topicName: tf
      3. 使用 sim timestamp
   ii. 里程计 Odometry: (**Isaac Compute Odometry Node + ROS2 Publish Odometry Node**)
      1. **Isaac Compute Odometry Node:** 配置 inputs:chassisPrim (articulation root prim)
      2. **ROS2 Publish Odometry Node:** 配置 chassisFrameId: base_link, odomFrameId: odom, topicName: odom
      3. 使用 sim timestamp

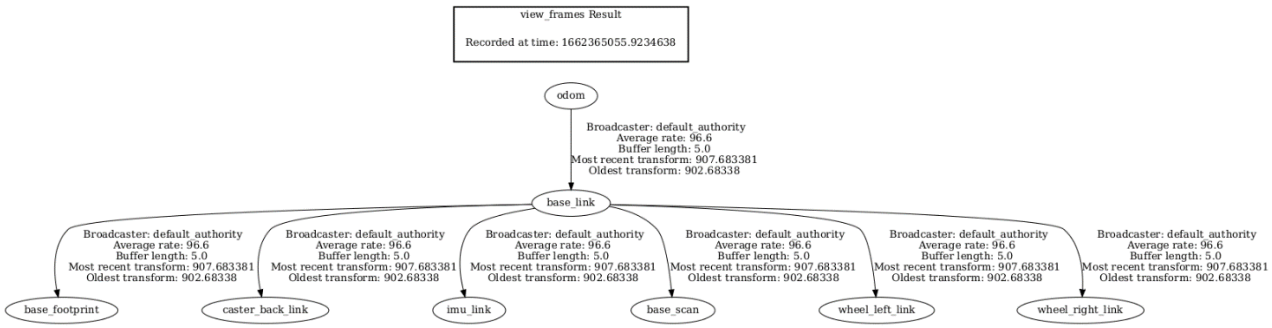
g. TF Tree: (**ROS2 Publish Transform Tree Node**)
   i. 配置 inputs:parentPrim, inputs:targetPrims
   ii. 关节树中所有的后继坐标框架都自动添加到 TF 中
      1. 确保添加的是 articulation root prim
   iii. 使用 sim time :



h. Sim time: /clock



4. 可视化 TF Tree: *ros2 run tf2_tools view_frames.py*



ROS2 案例编译安装

1. 安装必要的依赖库

 # For rosdep install command

sudo apt install python3-rosdep python3-rosinstall python3-rosinstall-generator python3-wstool build-essential

 # For colcon build command

sudo apt install python3-colcon-common-extensions

2. Source ROS2 安装环境

source /opt/ros/foxy/setup.bash

3. 解决依赖问题

cd ~/.local/share/ov/pkg/isaac_sim-2022.1.1/ros2_workspace

rosdep install -i --from-path src --rosdistro foxy –y

4. 编译工作空间

colcon build

5. Source 工作空间

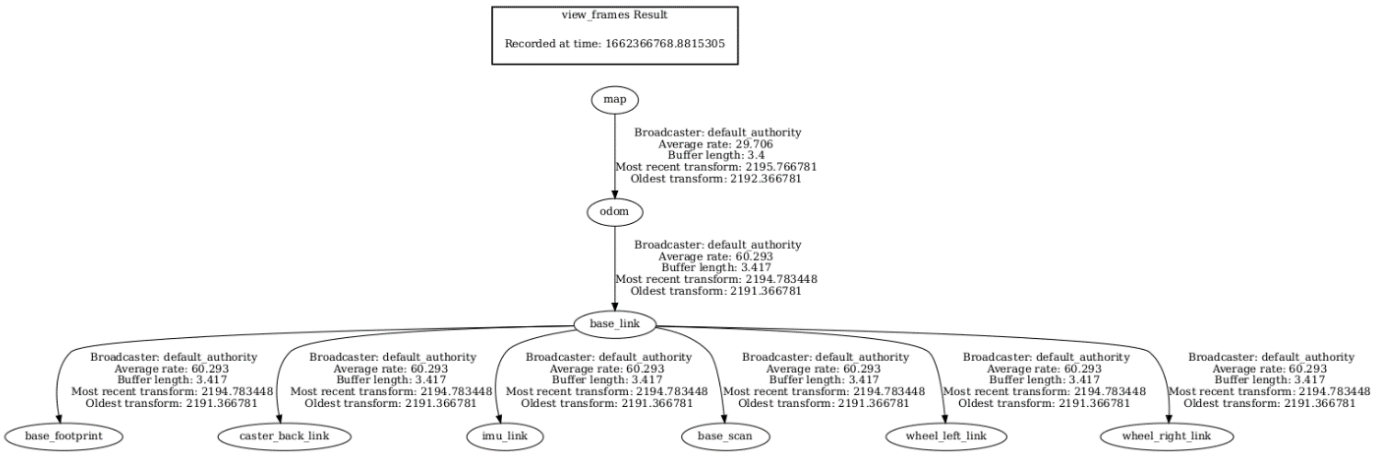
source install/local_setup.bash

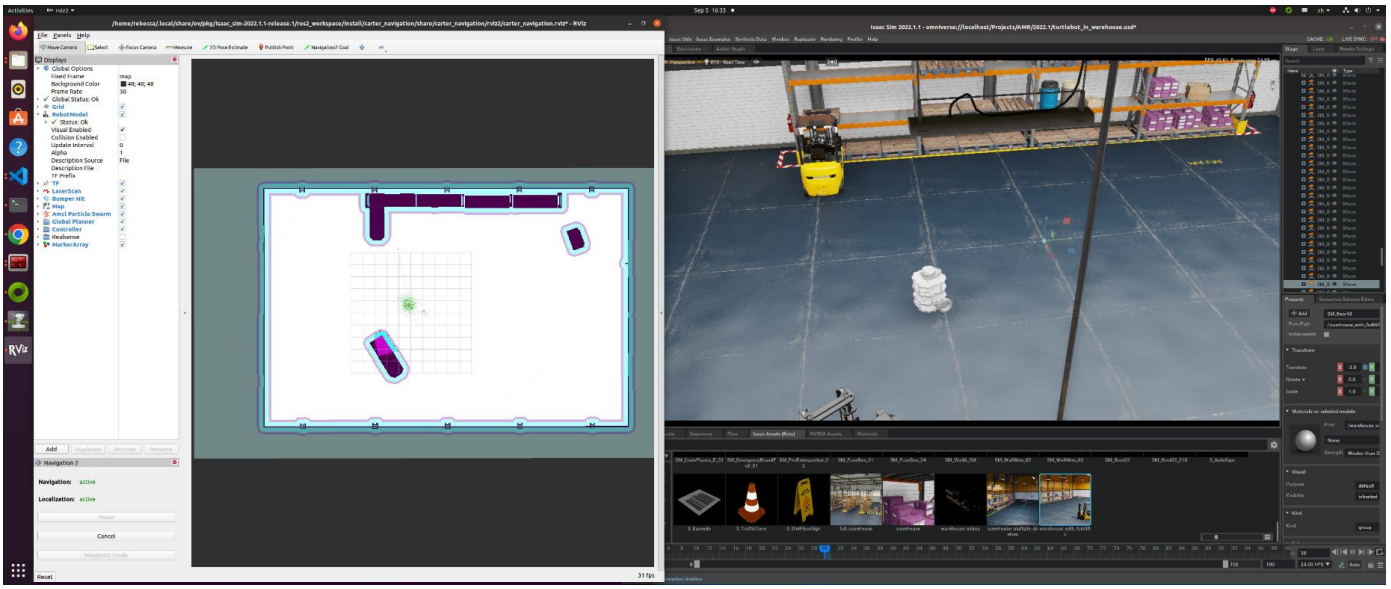
4. ROS2 Navigation 案例 [turtlebot_navigation_003]

1. 在配置好上述 ROS2 bridge OmniGraph 场景的 ISAAC SIM 中点击 PLAY 开始仿真；
2. 在上述 source 好 ros2 工作空间的 terminal 中运行：

ros2 launch carter_navigation carter_navigation.launch.py

4. 可视化 TF Tree: *ros2 run tf2_tools view_frames.py*




Activities
rviz2
/home/rebecca/local/share/ovplpkg/isaac_sim-2022.1.1-release.1/ros2_workspace/install/carter_navigation/share/carter_navigation/rviz2/carter_navigation.rviz* - RViz
File Panels Help
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
Publish Point
Navigation2 Goal
Displays
Global Options
Fixed Frame
map
Background Color
48; 48; 48
Frame Rate
30
Global Status: Ok
Grid
RobotModel
Status: Ok
Visual Enabled
Collision Enabled
Update Interval
0
Alpha
1
Description Source
File
Description File
TF Prefix
TF
LaserScan
Bumper Hit
Map
Amcl Particle Swarm
Global Planner
Controller
Realsense
MarkerArray
Add
Duplicate
Remove
Rename
Navigation 2
Navigation: active
Localization: active
Pause
Cancel
Waypoint mode
Reset
31 fps
Sep 5 16:33
Isaac Sim 2022.1.1 - omniverse://localhost/Projects/AMR/2022.1/turtlebot_in_warehouse.usd*
Isaac Utils
Isaac Examples
Synthetic Data
Window
Replicator
Rendering
Profiler
Help
CACHE: ON
LIVE SYNC: OFF
Stage
Layer
Render Settings
Search
Name
Type
Prim Path
Instanceable
Transform
Translate
Rotate
Scale
Materials on selected models
Prim
None
Strength
Visual
Purpose
default
Visibility
inherited
Kind
Kind
group
Isaac Assets (Beta)
NVIDIA Assets
Materials
S_Barrel01
S_TrafficCone
S_WetFloorSign
full_warehouse
warehouse
warehouse_nobox
warehouse_multiple_sh
ape
warehouse with forklift
Sequencer
Play
24.00 FPS