

ROSを試ってみよう！

Gr. 9 ロボティクスグループ
田窪朋仁

1

ROSとは？

- ROSはRobot Operating Systemの略
 - 現在はROS 2というバージョンでの開発が進んでいる
- オープンソースのロボットソフトウェアプラットフォームで、世界中のロボットアプリケーションの開発と実行に使用されている
- ノードベースのアーキテクチャを採用しており、異なるノードがメッセージをやり取りして協調動作

2

ROSの基本コンセプト

- パッケージ
 - ROSのコード、ライブラリ、設定ファイルなどをまとめた単位
- ノード
 - プログラムの単位で、特定の機能（タスク）を実行
 - 登録・変更・削除が容易にできる
- 通信方式
 - ノード間でデータを送受信するためのメッセージ通信の仕組みを採用
 - トピック・サービス・アクションと呼ばれる機能で通信

3

開発ツール

- プロジェクトの構成・ビルドシステム
 - C++, Pythonに対応
- GUIベースのデバッグツール
 - rqt, rvizと呼ばれるGUIによりグラフィカルにプログラムの状況を確認可能
- 3次元物理シミュレーション
 - Gazeboと呼ばれる3次元物理シミュレータと連携して実際のロボットをシミュレーション空間に再現してシミュレーション可能

4

汎用性のあるパッケージ群

- ロボットアームや移動ロボットの開発に必要な一般性のあるプログラムが公開されており自由に利用可能

- 代表的なパッケージ例

パッケージ名	機能
tf	ロボットの3次元座標と座標間のつながりを管理する
teleop	ロボットの移動方向の指示
MovelIt	ロボットアームの制御アルゴリズム
Navigation2	移動ロボットの地図作成・自己位置推定・経路計画アルゴリズム

5

ROSのメリットは？

- ロボットの研究開発でありがちなこと
 - プログラムが巨大になりがち
 - わかっている人が開発者本人だけとか...
 - イベントドリブンになりがち
 - 「センサAが反応したらBを実行」という似たようなパターンのプログラムをたくさん作る必要あり
 - 似ているようで微妙に違うとか...
- ROSを使うと？
 - 同じハードやアルゴリズムを使っている人が同じプラットフォームでプログラムを提供している
 - プログラムの部品化をすることで再利用・置き換え可能
 - ほかの人のプログラムを再利用することでゼロからプログラムを作らなくてよい
 - 部品のデバッグをすればデバッグ工数が大幅に減る

6

ROSはどこで学ぶ？

- ROS Wiki
 - <https://wiki.ros.org/ja>
 - ROSの文書・パッケージの説明が載っている
 - 情報量が多すぎるので初心者向けではない
- ROS 2チュートリアル
 - <https://docs.ros.org/en/foxy/Tutorials.html>
 - 英語ページですが自動翻訳でも大体わかる
 - まずはここから
- おすすめ書籍
 - ROS2とPythonで作って学ぶAIロボット入門
 - アマゾンで電子書籍もある
 - Windows上にDockerをインストールしてROS 2のプログラムを学習できる

7

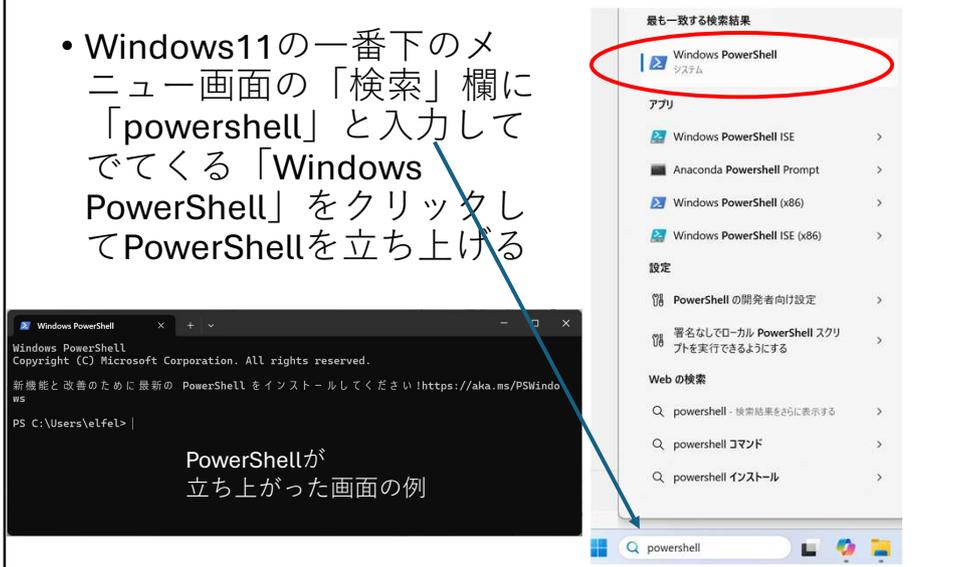
自宅でROS 2を使う方法

1. Windows11にWSL (Windows Subsystem for Linux)をインストール
2. Micro Soft StoreからUbuntu22.04.3LTSをインストール (無料です)
3. Ubuntu22.04.3LTSを実行して、その画面からROS 2をインストール

8

WindowsにWSLをインストール1/2

- Windows11の一番下のメニュー画面の「検索」欄に「powershell」と入力してでてくる「Windows PowerShell」をクリックしてPowerShellを立ち上げる



9

WindowsにWSLをインストール2/2

- PowerShellの画面で下記コマンドを入力してWSLをインストール

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

新機能と改善のために最新の PowerShell をインストールしてください!https://aka.ms/PSWindows

PS C:\Users\elfel> wsl.exe --install
```

10

WindowsにLinuxをインストール

- Windows11を持っている人は簡単にLinuxと呼ばれるOSを利用できます
- インストール方法
 - Microsoft Storeのアイコンをクリック
 - 出てきたウィンドウの上部に「ubuntu」と入力して検索
 - Ubuntu 22.04.3 LTS をインストール



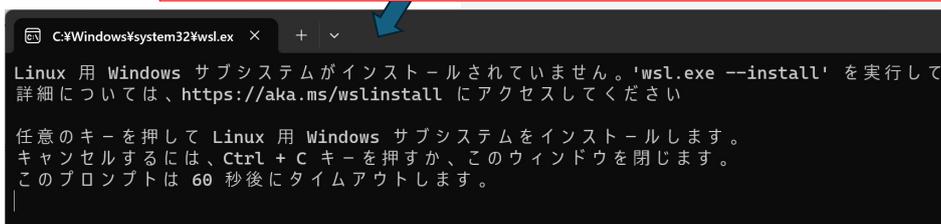
11

Linuxの起動

- インストール後にWindowsメニューから「すべてのアプリ」の一覧から「Ubuntu 22.04.3 LTS」を選択して起動
- 初回起動時は名前とパスワードを設定



クリック後にこの画面が出たら前のページを参考にWSLをインストール

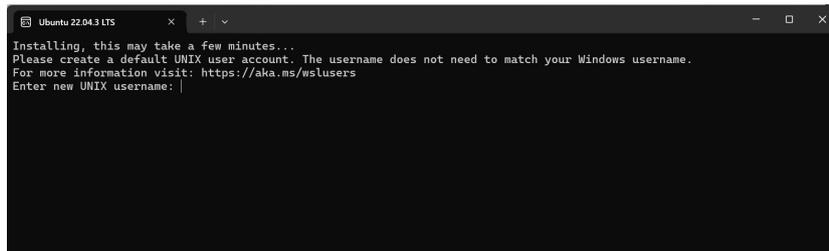


12

Linux初回起動時の画面

初めて起動したときは、下図のようにユーザ名
の入力が求められます。

好きなユーザ名を入力してください。
ユーザ名は半角の英語で入力すること

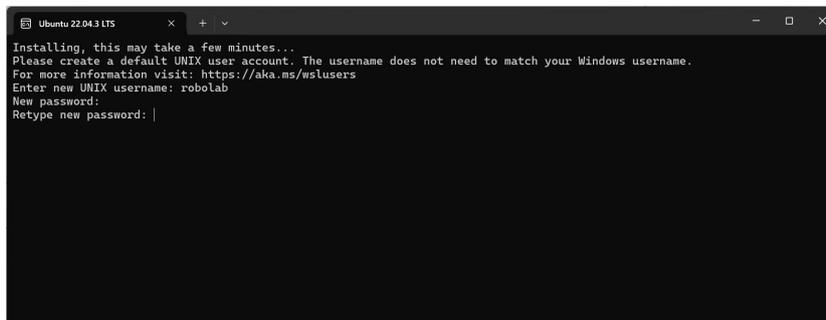


```
Ubuntu 22.04.3 LTS
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: |
```

13

パスワードの設定

下の画面のようにパスワードを2回入力し
て設定する

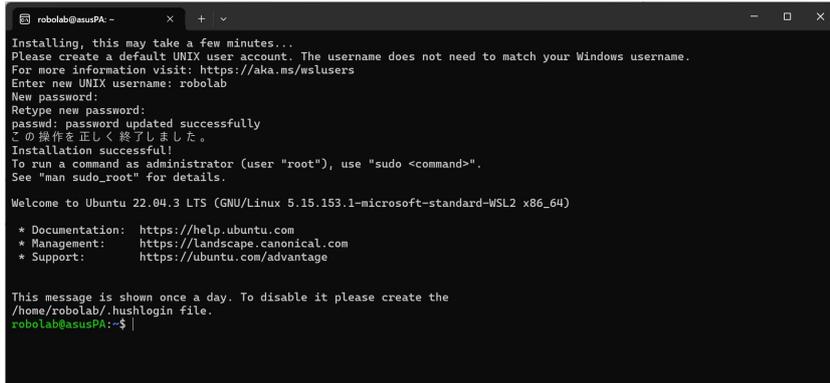


```
Ubuntu 22.04.3 LTS
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: robotlab
New password:
Retype new password: |
```

14

設定後の画面

パスワードの設定が終わると下記の画面のようになりLinuxの利用が可能になります。
2回目からはパスワードやユーザ名の入力なしにすぐに利用できる。



```
robotlab@ubuntu:~$
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: robotlab
New password:
Retype new password:
passwd: password updated successfully
この操作を正しく終了しました。
Installation successful!
To run * command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.153.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:   https://landscape.canonical.com
 * Support:      https://ubuntu.com/advantage

This message is shown once a day. To disable it please create the
/home/robotlab/.hushlogin file.
robotlab@ubuntu:~$
```

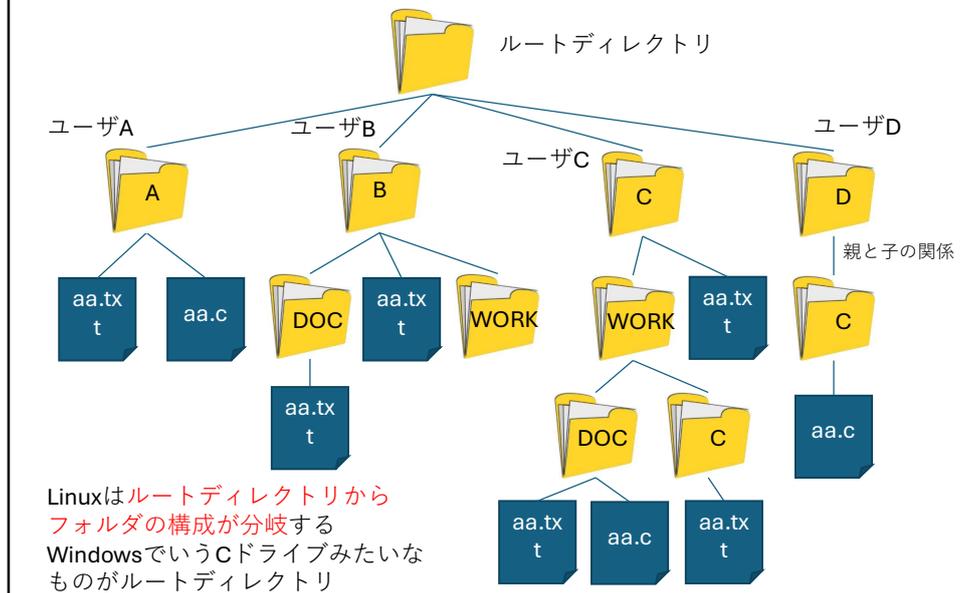
15

LinuxはCUI

- linuxはGUIでの操作はほとんどなく、CUIが基本となります
- CUIとは、「キャラクターユーザーインターフェイス」の略で、文字(Character)でコンピュータに指示を送る方式です
- Windowsでのフォルダの移動はGUIですが、Linuxではコマンドで移動します

16

ディレクトリ（フォルダ）の概要



17

パス名

- 階層型のディレクトリ構成を利用
 - ファイル名だけではデータにアクセスできない
 - パス名を使って解決
- パスとは？
 - ある基準からの経路
- 絶対パス（absolute path name）
 - ルートディレクトリからの経路
 - Ex. /C/WORK/DOC/aa.txt
- 相対パス（relative path name）
 - 作業ディレクトリorカレントディレクトリ（現在のディレクトリ）からの経路
 - 作業ディレクトリは、OSから与えられる
 - Ex. 作業ディレクトリ”B”の場合：
 - DOC/aa.txt (= ./DOC/aa.txt)
 - ../A/aa.txt
 - これらの相対パスのメリットは作業ディレクトリ”B”以下全体を動かしたときにパス名を変更する必要が無いことにある

18

ディレクトリの移動

- WindowsでのフォルダのことをLinuxではディレクトリと呼ぶ
- 現在いる（開いている）ディレクトリのことをカレントディレクトリと呼ぶ
- Linuxのホームディレクトリは「~」であらわすことができる
 - ホームディレクトリはWindowsでいう「c:¥ユーザー ¥ユーザ名」のフォルダのこと
- Linuxのディレクトリの移動は「cd」コマンドで実行する

19

Linux コマンド例

- cd コマンド（移動コマンド）
 - cdの後に半角スペースを入力後にパス名を記載
 - 例「cd /home/ユーザ名」と記載してエンター
 - このコマンドでホームディレクトリへ移動できる
 - cd~ このコマンドだけでもホームディレクトリへ移動可
- ディレクトリの作成
 - 「mkdir ディレクトリ名」エンターで作成
 - 例：mkdir ros2_ws
 - 現在いるディレクトリにros2_wsというディレクトリを作成
- 現在のディレクトリのリストを確認
 - ls コマンド
 - 「ls」と入力すると現在いるディレクトリの中にあるファイルとディレクトリの名前一覧を表示
 - 「ls -al」と入力すると名前だけでなく、ファイルごとの詳細な情報一覧を表示

20

ROS 2のインストール

- Linuxを立ち上げて出てきたウィンドウに下記のコマンドを順番に入力
 - `sudo apt install software-properties-common`
 - `sudo add-apt-repository universe`
 - `sudo apt update && sudo apt install curl`
 - `sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg`
 - `echo "deb [arch=$(dpkg --print-architecture)] signed-by=/usr/share/keyrings/ros-archive-keyring.gpg http://packages.ros.org/ros2/ubuntu $(. /etc/os-release && echo $UBUNTU_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null`
 - `sudo apt update && sudo apt upgrade`
 - `sudo apt install ros-humble-desktop`

書き写すのが大変なので、下記サイトにアクセスしてコピー & ペーストでOK
<https://sites.google.com/site/robotlabo/ros-2関連/ros-2のインストールhumble>

21

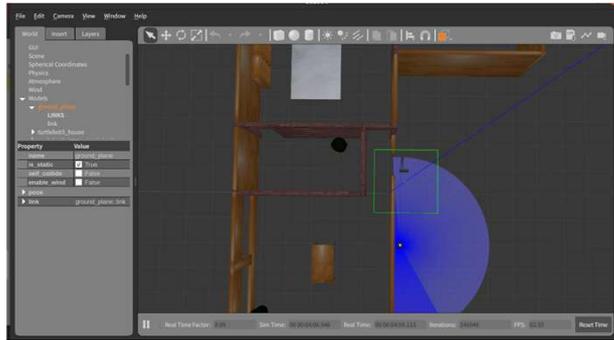
本日のROS 2プログラム体験

- 下記の2つの課題を各自のノートパソコンで実行
 1. 仮想環境上にある移動ロボットを操作してロボットに搭載されたレーザセンサで地図を製作
 2. 作成した地図を読み込んで地図上の任意の位置へ自律走行させる

22

遠隔操作による地図作成1/3

- コマンドプロンプトに下記コマンド入力
- `ros2 launch turtlebot3_gazebo turtlebot3_house.launch.py`
- 下記画面の物理シミュレータが立ち上がる

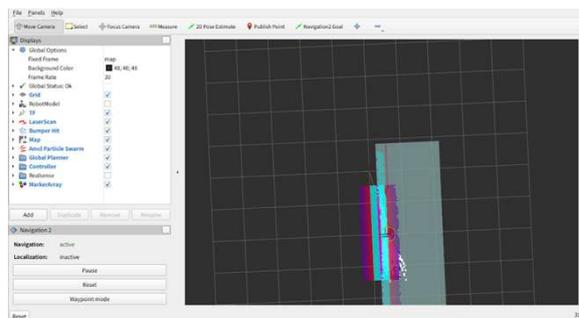


23

遠隔操作による地図作成2/3

- もう一つのコマンドプロンプトで下記コマンドを実行
- `ros2 launch turtlebot3_navigation2 navigation2.launch.py use_sim_time:=True slam:=True`

右のような地図作成のGUI画面が立ち上がる



24

遠隔操作による地図作成3/3

- もう一つのコマンドプロンプトで下記コマンドを実行
- `ros2 run turtlebot3_teleop teleop_keyboard`
- コマンドプロンプト画面が下記のようなになる

```

Control Your TurtleBot3!
-----
Moving around:
  w
  a  s  d
  x

w/x : increase/decrease linear velocity (Burger : ~ 0.22, Waffle and Waffle PI : ~ 0.26)
a/d : increase/decrease angular velocity (Burger : ~ 2.84, Waffle and Waffle PI : ~ 1.82)

space key, s : force stop

CTRL-C to quit
  
```

この画面記載の通りキーボードでロボットを操作ができる

w: 前進
a: 左旋回
d: 右旋回
x: 後退
s: ストップ
ctl+cを入力するとプログラム停止

25

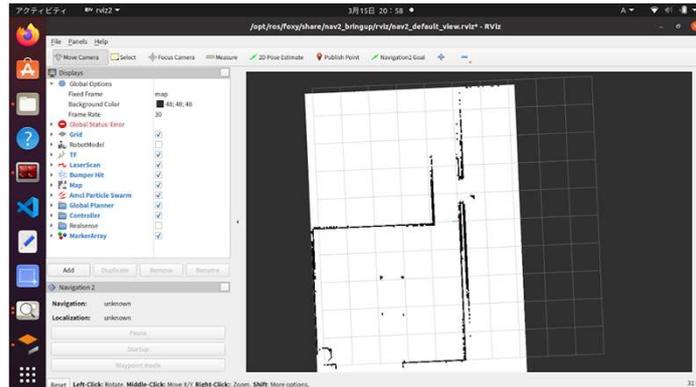
地図を保存と再利用

- 新しいコマンドプロンプトで下記コマンドを入力するとレーザセンサで作成した地図をコマンドを実行したカレントディレクトリに保存できます
- `ros2 run nav2_map_server map_saver_cli -f test_map`
- 保存した地図は下記コマンドでロボットが読み取ることができます
- `ros2 launch turtlebot3_navigation2 navigation2.launch.py use_sim_time:=True map:=$HOME/tmp/test_map.yaml`
 - `$HOME/tmp/test_map.yaml`は保存したファイルのパスを記載

26

地図上での自律移動1/5

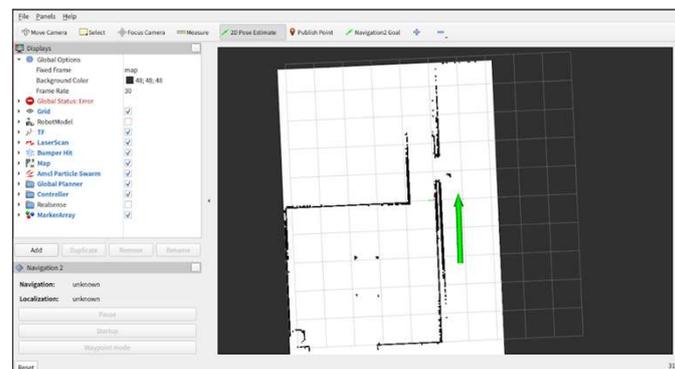
- 地図を読み込んだ直後の状態
- 地図を読み込んだだけでその地図のどこにロボットが位置しているかわかっていない



27

地図上での自律移動2/5

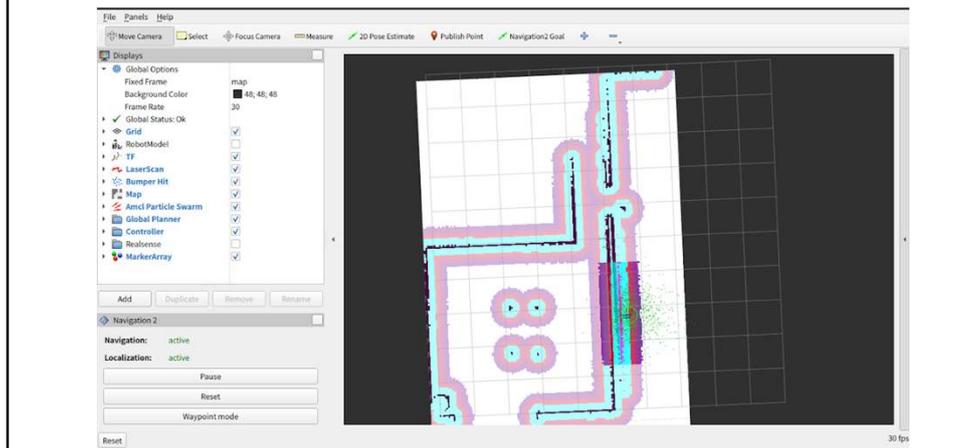
- 「2D pose Estimate」というボタンを押してから、ロボットが現在そのような地図の場所にドラッグして方向を矢印で表示させて後にマウスボタンを離す



28

地図上での自律移動3/5

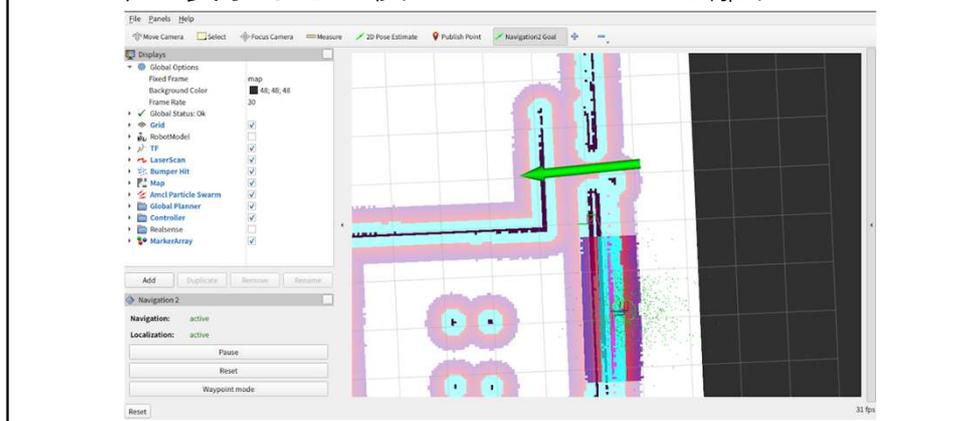
- ロボットがいるかもしれない場所に小さな点が複数表示される
 - ロボット自身が推定している地図上の場所の候補



29

地図上での自律移動4/5

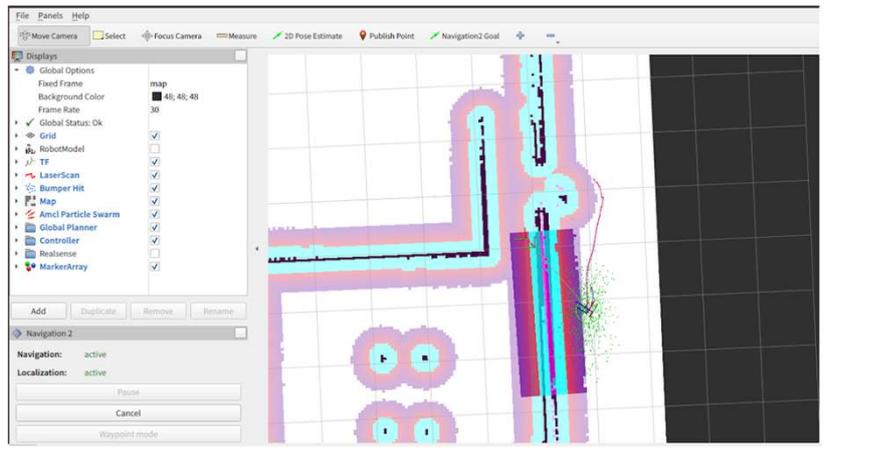
- 目標位置の設定
- 「Navigation2 Goal」というボタンを押してから、地図上の目標位置にドラッグして方向を矢印で表示させて後にマウスボタンを離す



30

地図上での自律移動5/5

- 指定した目標位置までの経路が計画され自律的に目標位置へ移動を始める



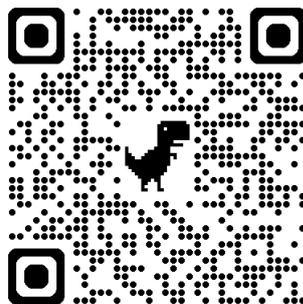
31

参考URL

- 大阪公立大学 電気電子システム工学科
ロボティクスグループ

ROS2関連ページ

<https://sites.google.com/site/robotlabo/ros-2関連>



32