

抜粋版

はじめてのIsaac-sim学習・開発キット ～フィジカルAI実践し、ロボット導入前に検証・確認。コスパに優れる～ 開発編 (ubuntu22.04版)



スペクトラム・テクノロジー株式会社

<https://spectrum-tech.co.jp>

sales1@spectrum-tech.co.jp

開発編 目次

• isaac-sim学習・開発キット 全体像	<u>3</u>
• ハード・ソフトウェア概要	
1. ハード概要	
① ハード仕様	<u>4</u>
② USB読み込み	<u>5</u>
2. ソフト概要	
① ソフト一覧	<u>6</u>
② プログラム一覧	<u>7</u>
1. isaac-simとは	<u>8</u>
2. isaac-sim概要	<u>9</u>
3. isaac-sim開発デモ	<u>18</u>
4. isaac-sim開発実践	<u>24</u>

isaac-sim 学習・開発キット 全体像

ハードウェア

CPU



32GB RAM

GPU



+

USBメモリ



OS

ubuntu

アプリ関係

Robot arm

Navi

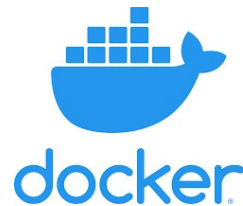
Digital twin

isaac系



NVIDIA

Isaac-sim



ROS2系



MoveIt

プログラム言語

python



お客様準備

弊社提供

1. ハード概要

①. 必要なハードウェア仕様

開発キットと必要なハードウェアの概要です。

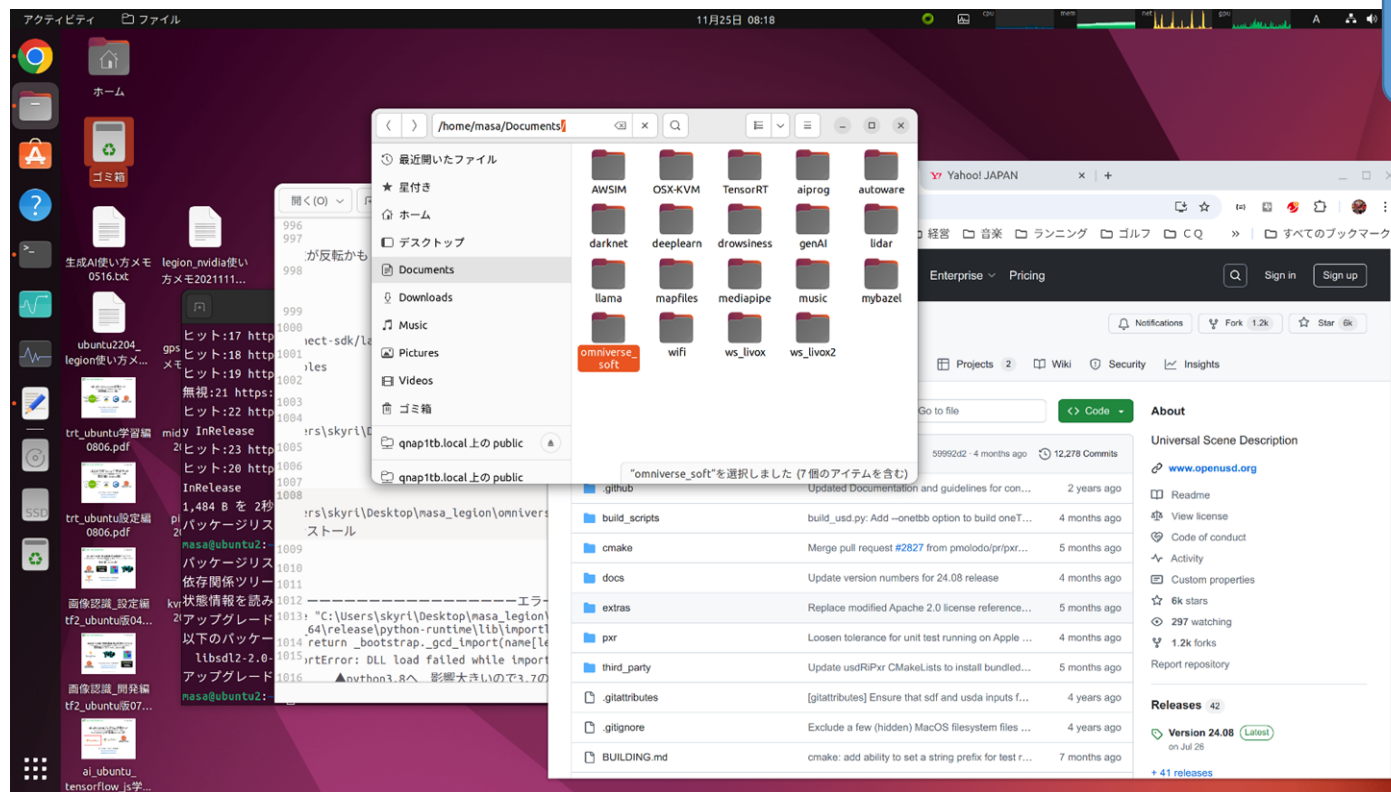
品名		仕様	備考
USBメモリ(ubuntu版)		USB3.0 64GB isaac-simプログラム一式	
必要なハードウェア:お客様準備			
PC本体	cpu	Intel i7 gen 7 AMD Ryzen 5	
	メモリ	32GB以上	
	Disk	500GB以上	
	GPU	RTX4070以上	

1. ハード概要

② USBのデータ読込

- 添付のUSBのデータを読込ます。
- 例: `cd /home/omni/Documents`配下にインポート/omniverse_soft

お客様毎に異なります



2. ソフト概要

①ソフトウェア一覧

本PCにインストールするソフトウェアの概要です。

区分	ソフト名	バージョン	備考
お客様準備: 弊社インストール・サービスにより提供可能			
OS	ubuntu	22.04 LTS	
GPU driver		580	
USBメモリにより提供: 手順書に基づきお客様でインストールして頂きます			
開発ツール	gcc	V11.4以上	
	cmake	V3.24	
	docker	29.7	
	ros2	humble	
プログラム言語	python3	3.10以上	
isaac-sim	isaac-sim	6.0.0	
アプリ	isaac-sim-ros, moveit, robot arm, lidar, camなど多数		

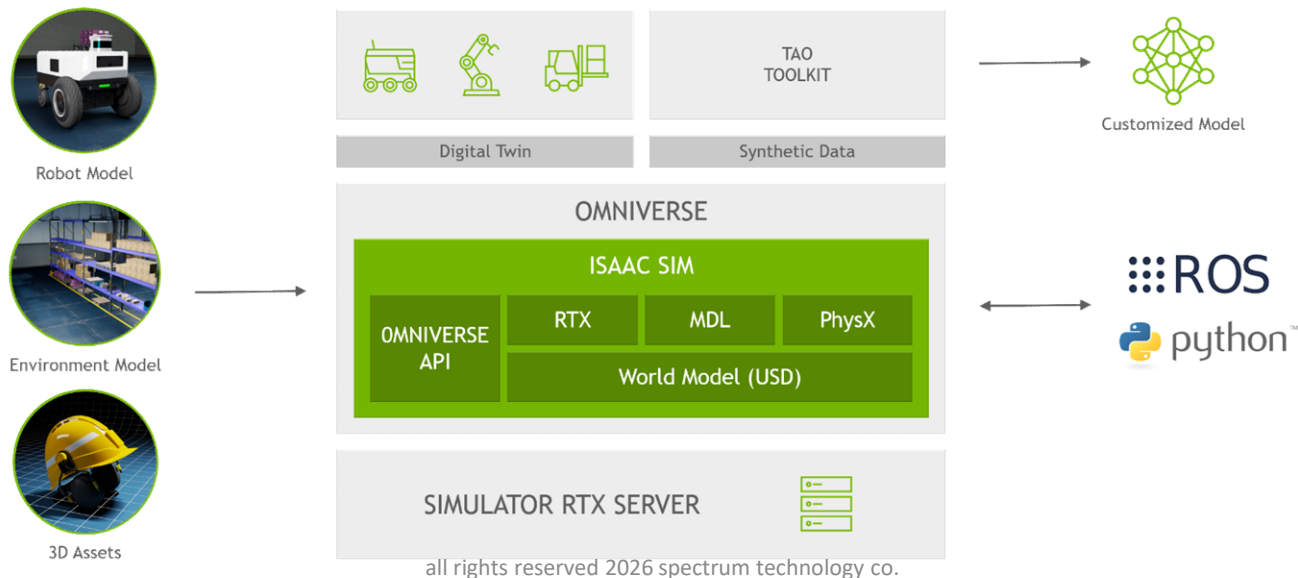
2. ソフトウェア概要

② 開発キットプログラム一覧

id	区分	項目	概要	難易度	コメント
1	isaac-sim概要	画面	Isaac simの画面です。openusd編で内容はよく確認してください	C	
2		構成	ジオメトリ作成: 3D CAD, ロボットなど インポート: 他のシステムから部品などをインポートできます。 Openusd: 基本は、こちらを使います シーン作成: 物理状態、センサ、照明などを設定 関連操作: python, グラフなどを使ってros2と連携させて、動作させます。	B	
3		資産構成	エンド・エフェクタ: ロボットハンド、吸盤。 制御: センサ、ROS連携 物理: 物理要素 資産: 対象物品など	B	
4		基本アプリ	Isaac lab	B	
5			ros2	A	
6		開発ツール	script editor	B	
7		ロボット・センサ設定	カメラ	B	
8			Lidar	A	
9			physics	A	
10	isaac-sim開発デモ	isaac-simを起動	dockerによる起動	B	
11		nova carter	Nova Carterは、次世代の自律走行搬送ロボット(AMR)の開発と導入を加速させる、包括的なロボティクス開発プラットフォームです	B	
12		iw_hub	倉庫内の自走ロボットを動作させるデモ	B	
13		turtlebot	turtlebotでカメラ画像を取得するデモ	C	
14		moveit2 pick&placeデモ1	Moveit2を使って物体をpick&placeするデモ。Ros2単独	B	
15		moveit2 pick&placeデモ2	Moveit2を使って物体をpick&placeするデモ。Ros2-Isaac sim連携	A	
16	isaac-sim開発実践	物品持ち上げ、横棒貫通	個別プログラムを使って物体(10cm位)を持ち上げて、約6mmの穴に横棒(3mm)を通す。	A	
17		横棒貫通テスト	個別プログラムを使って物体(10cm位)を横棒(3mm)を通す	C	
18		物体pickup	個別プログラムを使って物体(10cm位)を持ち上げる	B	

1. isaac-simとは

- NVIDIA Isaac Sim は、Omniverse を基盤とするスケーラブルなロボティクス シミュレーション アプリケーション、および合成データ生成ツールです。現場を忠実に再現したフォトリアリスティックな仮想環境は、AI ベースのロボットの開発、テスト、管理に最適です。
- 特徴は、実物のロボットを購入しなくてもシミュレーションでき、ros2などのプログラムはそのまま実機と連携が可能
- <https://docs.isaacsim.omniverse.nvidia.com/6.0.0/index.html>



2.isaac-sim概要

(1)画面

https://docs.isaacsim.omniverse.nvidia.com/6.0.0/gui/reference_user_interface.html

Isaac simの画面です。openusd編で内容はよく確認してください。

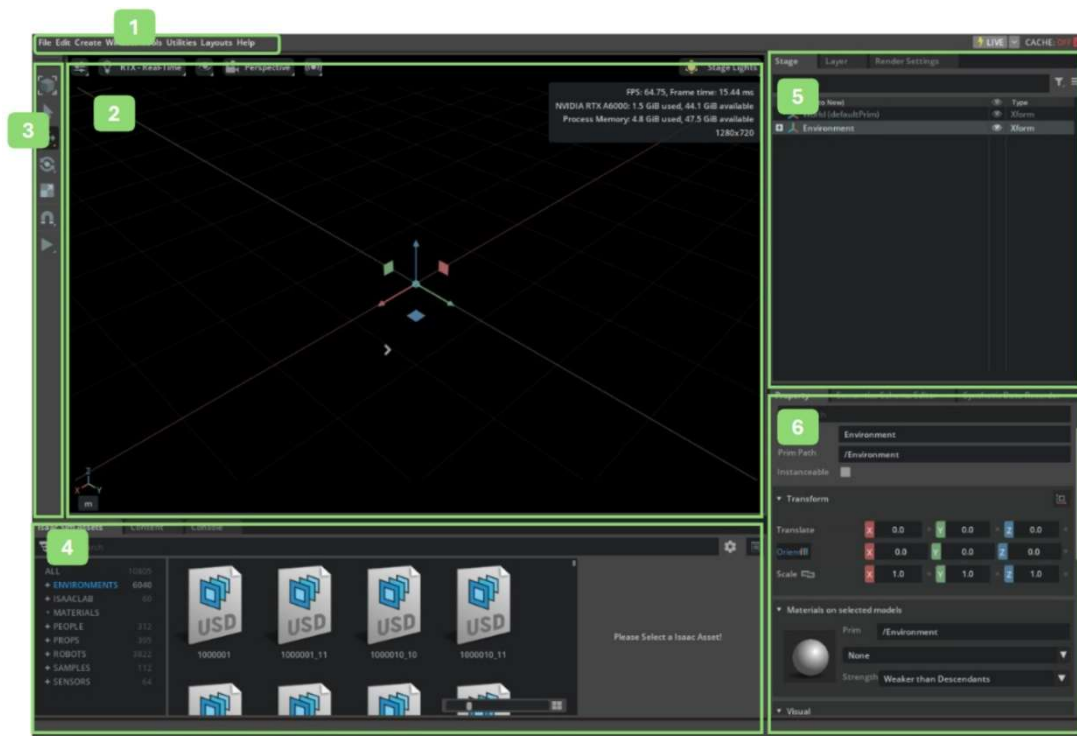
起動

isaac-sim@ubuntu2:~\$

./runapp.sh

起動方法は、[こちら](#)

画面



項番	UI要素名	日本語解説
1	Menu Bar (Isaac Sim Menu Bar)	各種設定、ウィンドウの表示切り替え、ファイル操作、拡張機能の管理などを行うメニューバーです。
2	Viewport	3Dシーンやアセットを表示・確認するためのメイン画面(ビューポート)です。
3	Main Toolbar	アセットの移動・回転・拡大縮小などの操作ツールや、シミュレーションの開始・停止ボタンが配置されているツールバーです。
4	Browsers	アセットやサンプルプロジェクトをブラウズ(検索・選択)するためのデフォルトの領域です。
5	Stage	現在のUSDシーンに含まれるすべてのオブジェクト(プライム)の階層構造を確認・管理できるステージウィンドウです。
6	Property Panel	選択したプライム(オブジェクト)の詳細なプロパティ(属性や物理設定など)を表示・編集するウィンドウです。

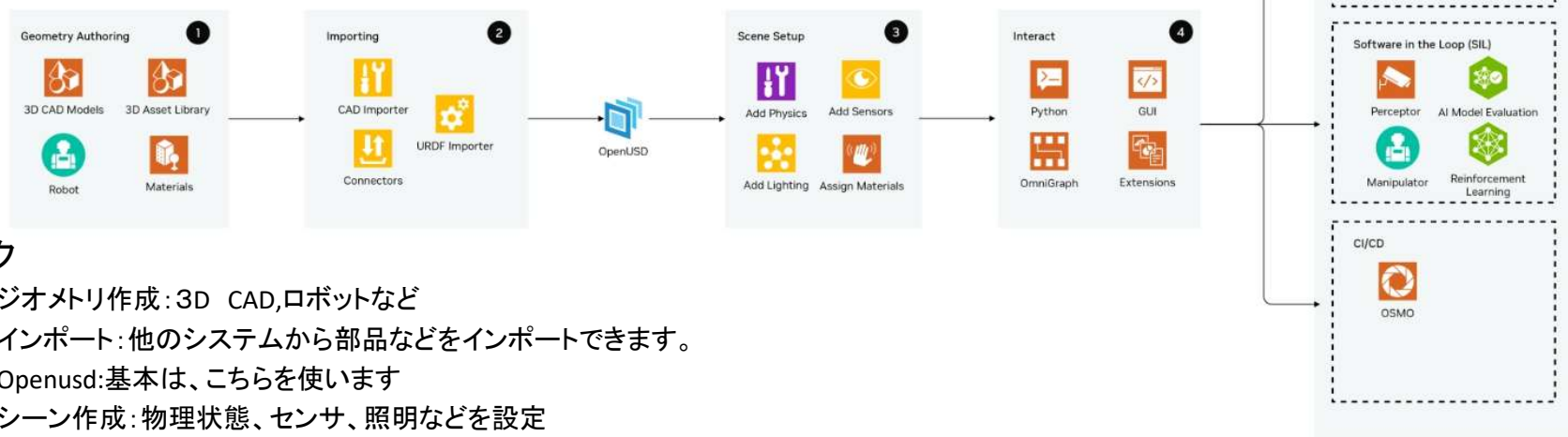
2.isaac-sim概要

(2)構成

https://docs.isaacsim.omniverse.nvidia.com/6.0.0/introduction/reference_architecture.html

Isaac simの構成です。

構成図



タスク

ジオメトリ作成: 3D CAD, ロボットなど

インポート: 他のシステムから部品などをインポートできます。

Openusd: 基本は、こちらを使います

シーン作成: 物理状態、センサ、照明などを設定

関連操作: python, グラフなどを使ってros2と連携させて、動作させます。

2.isaac-sim概要

(4) 基本アプリ

① Isaac lab

https://docs.isaacsim.omniverse.nvidia.com/6.0.0/isaac_lab_tutorials/index.html

Isaac labの内容。

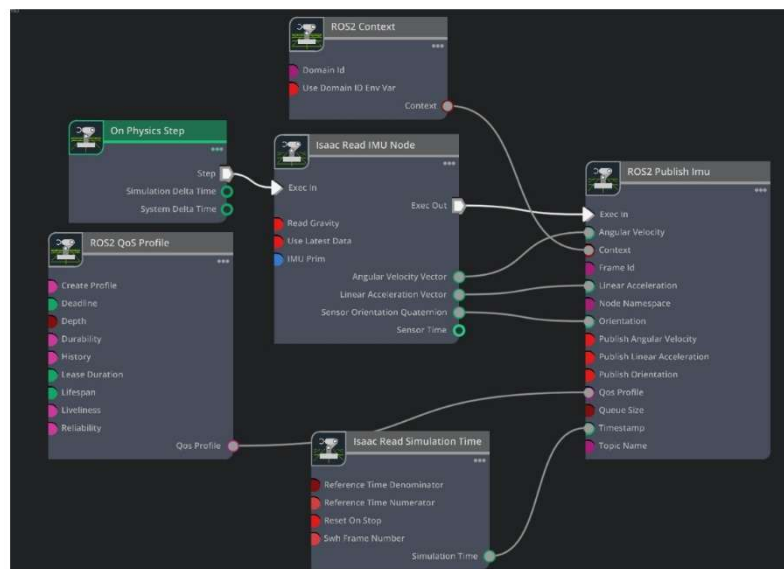
ロボット例

- Window>example:Robot exampleに多数の例。ロボットアーム、humanoidなど

Ros連携

Ros2 ブリッジ: Window > Extensionsでros2 bridgeをonとします。必須

Actiongraph>window>graph editorで選択。下の方に表示。Ros2のtopic 送受信などをこちらで設定。作成方法は別途



2.isaac-sim概要

(6)ロボット・センサ設定

③physics

<https://docs.isaacsim.omniverse.nvidia.com/6.0.0/physics/index.html>

Isacc simの物理の設定。sim独自のもので、この設定により、落下、摩擦、材質などを変更することができる。

設定

Rigid Body

物体を選んで、Add > Physics > Rigid Body: かたい物体になり、落下することができる。

Collider

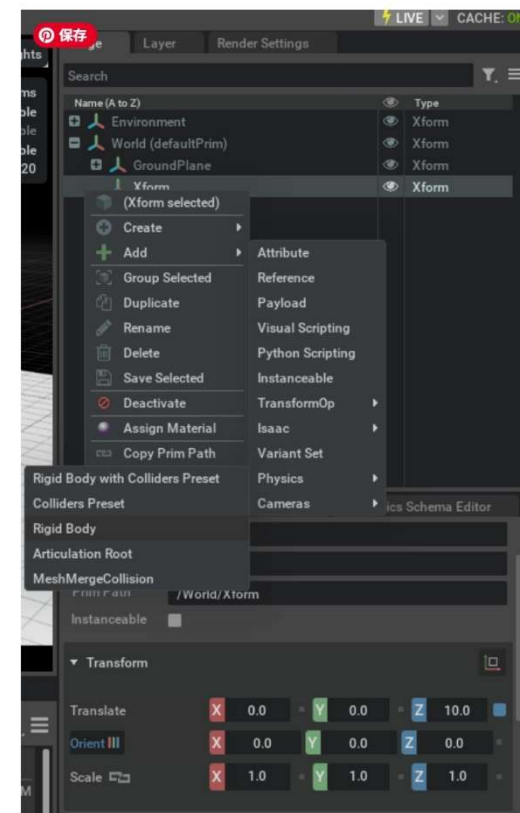
Add Button > Physics > Collider 衝突した時の状態を設定

動作確認

Playで、物体が空中にあれば、落下します。

その他

Kinematic:シミュレーション中に落下などを停止することができます。よく使います。



3.isaac-sim開発デモ

(1)nova carter

https://docs.isaacsim.omniverse.nvidia.com/6.0.0/assets/nova_carter_landing_page.html?highlight=nova#id1

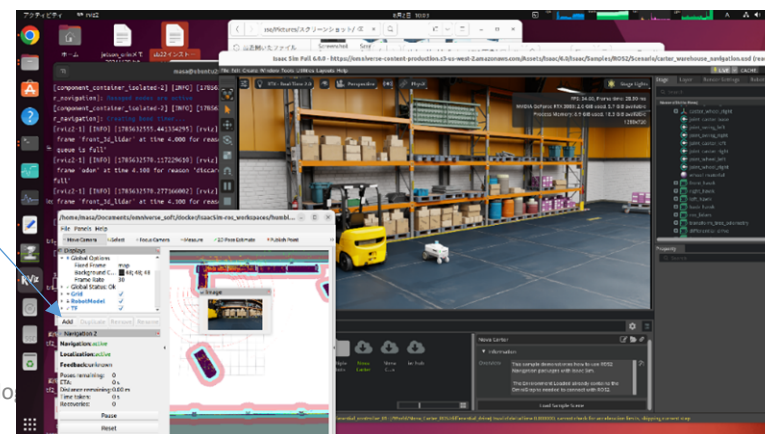
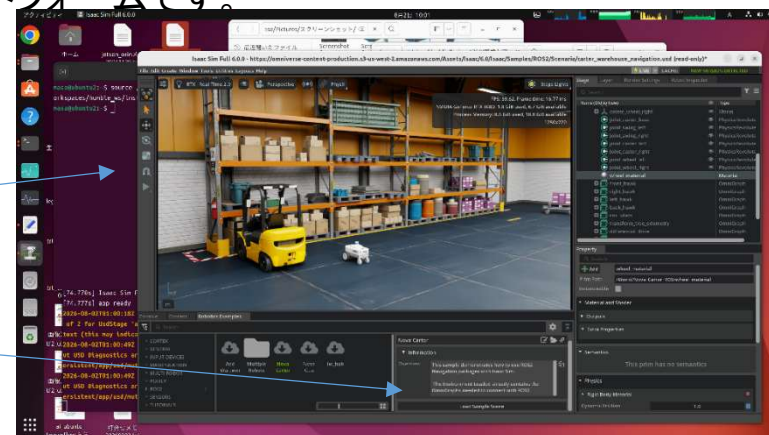
Nova Orin™センサーおよびコンピュータ・アーキテクチャを搭載したNova Carterは、次世代の自律走行搬送ロボット(AMR)の開発と導入を加速させる、包括的なロボティクス開発プラットフォームです。

起動

```
isaac-sim@ubuntu2:~$ ./runapp.sh
```

テスト

- Isaac-simを立ち上げて
 - Window ➤ Examples > Robotics Examples
 - ROS2 > Navigation > Nova Carter ロード
 - play
 - 別端末で以下を
 - source /home/omni/Documents/omniverse_soft/docker/IsaacSim-ros_workspaces/humble_ws/install/setup.bash 毎回実施
 - ros2 launch carter_navigation carter_navigation.launch.py
- rviz2が起動して、lidarなどが動いている様子が表示



ST Spectrum Technology

3.isaac-sim開発デモ

(2) iw_hub

https://docs.isaacsim.omniverse.nvidia.com/6.0.0/ros2_tutorials/tutorial_ros2_navigation.html?highlight=iw#nav-2-with-iw-hub-in-warehouse

倉庫内の自走ロボットを動作させるデモ。

起動

```
isaac-sim@ubuntu2:~$ ./runapp.sh
```

テスト

➤ Isaac-simを立ち上げて

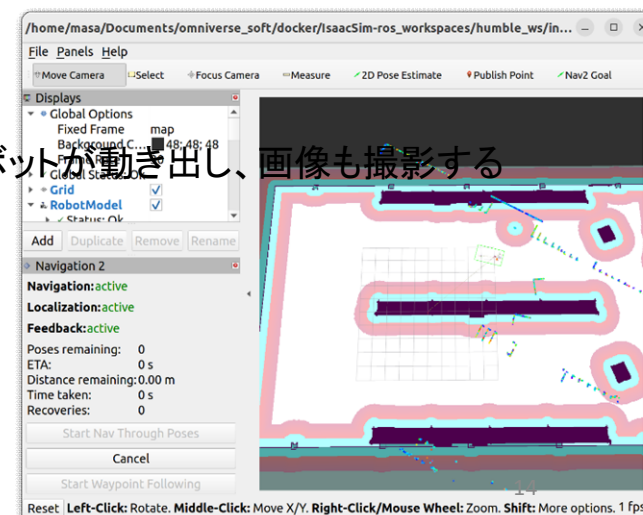
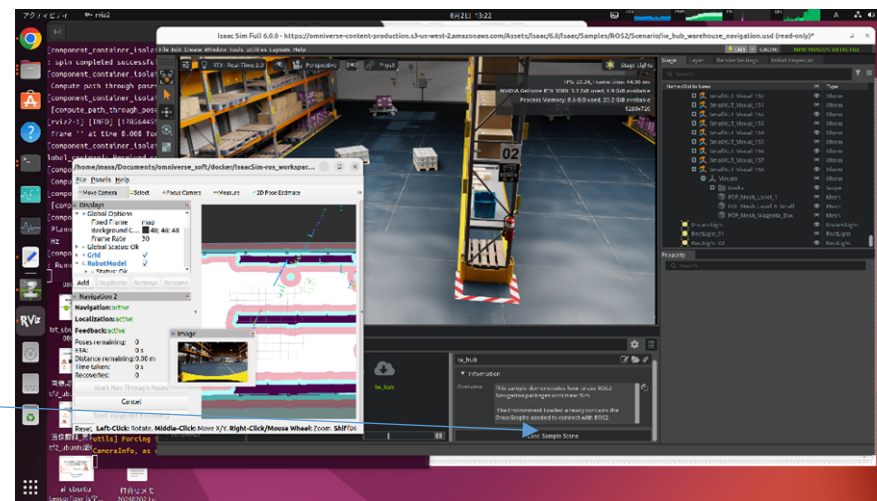
- Window > Examples > Robotics Examples
- ROS2 > Navigation > iw_hub ロード
- play

➤ 別端末で以下を

➤ `source /home/omni/Documents/omniverse_soft/docker/IsaacSim-ros_workspaces/humble_ws/install/setup.bash` 毎回実施

➤ `ros2 launch iw_hub_navigation iw_hub_navigation.launch.py`

上の2d estimationで目標位置を決め、左下のNAVをスタートさせるとロボットが動き出し、画像も撮影する



ST Spectrum Technology

3.isaac-sim開発デモ

(4) moveit2 pick&placeデモ1

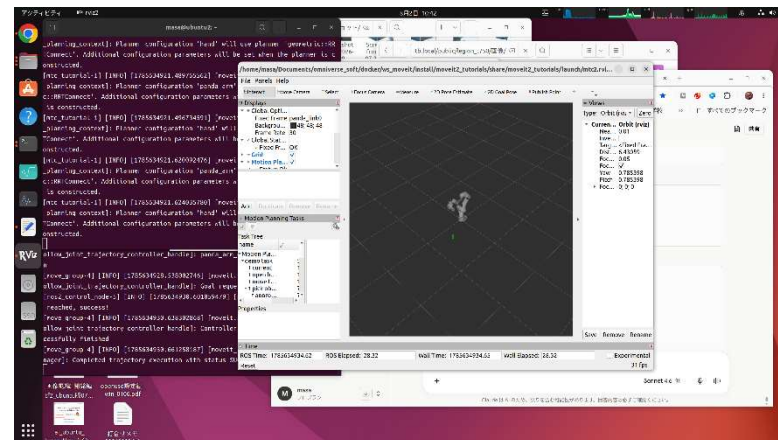
Moveit2を使って物体をpick&placeするデモ。Ros2単独

<https://moveit.picknik.ai/main/index.html>

moveit task constructorを利用し、タスクを自動化

起動

- `source /home/omni/Documents/omniverse_soft/docker/ws_moveit/install/setup.bash`
- `ros2 launch moveit2_tutorials mtc_demo.launch2.py`
- 別端末で
- `ros2 launch moveit2_tutorials pick_place_demo.launch.py`



ST Spectrum Technology

4.isaac-sim開発実践

(1)物品持ち上げ、横棒にさす。難易度A

個別プログラムを使って物体(10cm位)を持ち上げて、約6mmの穴に横棒(3mm)を通す。

構成

- 2個の物体を向きを変え、反転させたりしている。
- Ur10 ロボットアーム、吸着盤
- Lidar(livox mid70など想定)、物品の初期位置を把握
- カメラ: 横棒の近くに据え付けたカメラで穴の位置を把握

起動

Isaac sim:open>stacking_loop_lidar_cam9.usd

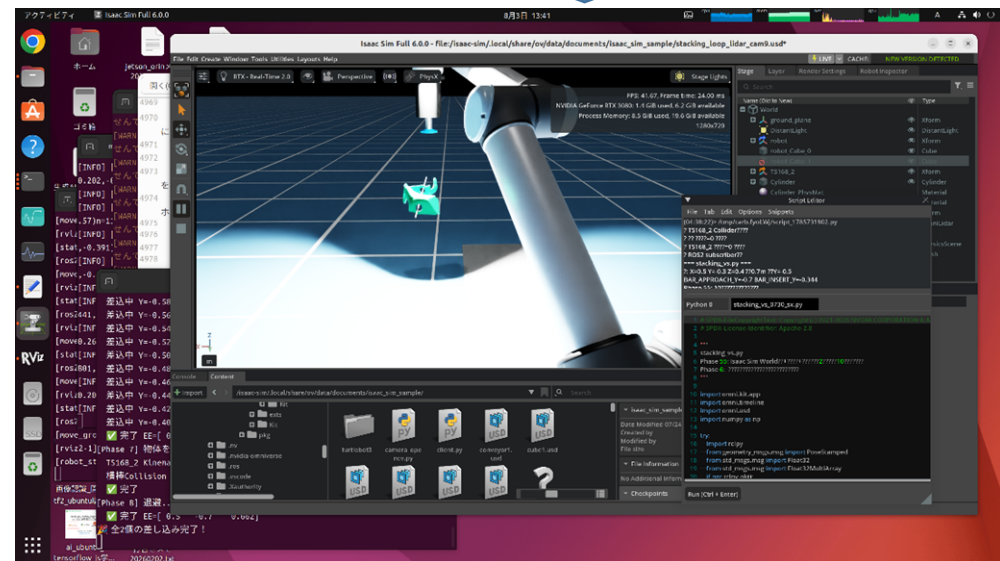
- Script editor: ./isaac-sim/.local/share/ov/data/documents/stacking_vs_0730_sx.py
- 下の3つのプログラムを起動して、run,play

Kinematic:onで終了。次にplayするときはoffへ。

動作

端末に動作状態を表示、右は2個目を貫通

```
isaac-sim@ubuntu2: ~
[Phase 0] 上空へ [ 0.26885104 -0.44016076 0.3 ]
Module isaacsim.core.experimental.utils.impl.transform ae47883 load on device 'cuda:0' took 2.25 ms (cached)
Module isaacsim.core.experimental.utils.impl.transform e15368d load on device 'cuda:0' took 2.37 ms (cached)
[Phase 1] 静止...
[Phase 2] 下降...
[Phase 3] 吸着...
[Phase 4] リフトアップ...
完了 EE=[ 0.269 -0.44 0.549]
通常配置(orient=-89.5): PCA_OFFSET=181.8 →rotateZ=66.5度
PCA: -115.3度 →回転=66.5度
✅ TS168_2 z軸66.5度回転 (PCA)
[Phase 45] 回転後安定待機...
[Phase 5] 手前へ... [ 0.5 -0.7 0.55]
EE=[ 0.498 -0.696 0.526]
EE=[ 0.5 -0.7 0.55]
EE=[ 0.5 -0.7 0.55]
EE=[ 0.5 -0.7 0.55]
```



ST Spectrum Technology

4.isaac-sim開発実践

(2)横棒貫通テスト。難易度C

個別プログラムを使って物体(10cm位)を横棒(3mm)を通す。

構成

- 1個の物体。貫通後kine:offにして横棒にぶら下がる

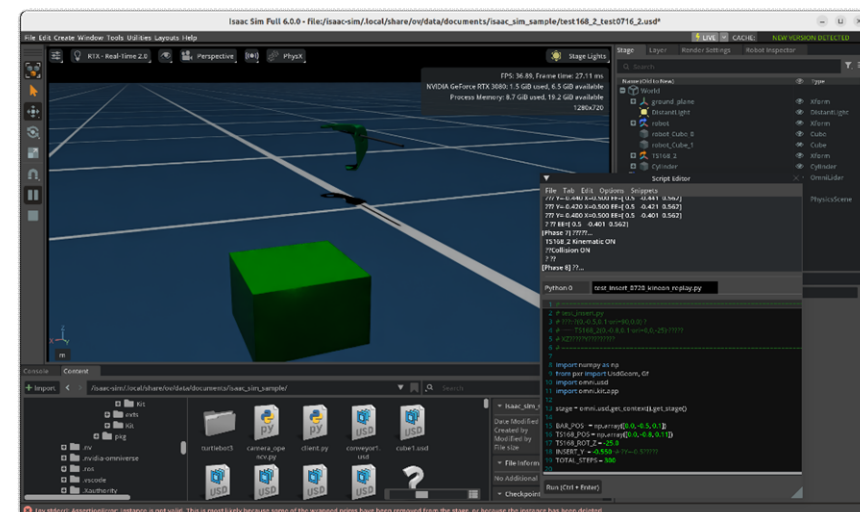
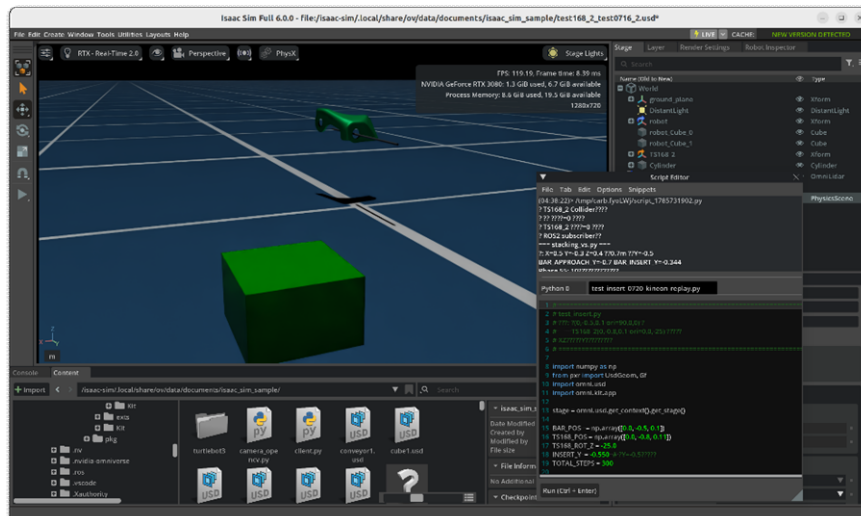
起動

Isaac sim:open>ts168_2_test0716_2.usd

Script editor: ./isaac-sim/.local/share/ov/data/documents/test_insert_0720_kineon_replay.py
run,play

コメント

- 右がテスト完了後、kinematic:offとして横棒にぶら下がった状態。
- 通常は、物体と横棒でcollision発生して、はじかれてしまう。



4.isaac-sim開発実践

(3) 物体pickup。難易度B

個別プログラムを使って物体(10cm位)を持ち上げる。

構成

- 1個の物体。持ち上げる
- Panda arm
- Ros2 mtc

起動

Isaac sim:open>panda_2_livox5.usd

下記プログラムを動作後に、play

- source /home/omni/Documents/omniverse_soft/docker/ws_moveit/install/setup.bash
- ros2 launch moveit2_tutorials demo.launch.py rviz_config:=panda_hello_moveit.rviz use_sim_time:=true
- 別端末で、プログラムでつかむ
- cd /home/omni/Documents/omniverse_soft/docker/ws_moveit
- python3 grasp_demo2.py
- 別端末で、メッセージ変換
- cd /home/omni/Documents/omniverse_soft/docker/ws_moveit
- python3 bridge2.py

コメント

- Panda armのgripperだとつかめない。

