

# 22、ROS2 Gazebo仿真平台

## 1、Gazebo简介

Gazebo是ROS系统中最为常用的**三维物理仿真平台**，支持动力学引擎，可以实现高质量的图形渲染，不仅可以模拟机器人及周边环境，还可以加入摩擦力、弹性系数等物理属性。

比如我们要开发一个火星车，那就可以在Gazebo中模拟火星表面的环境，再比如我们做无人机，续航和限飞都导致我们没有办法频繁用实物做实验，此时不妨使用Gazebo先做仿真，等算法开发的差不多了，再部署到实物上来运行。

所以类似Gazebo这样的仿真平台，可以帮助我们验证机器人算法、优化机器人设计、测试机器人场景应用，为机器人开发提供更多可能。

**注意：本章节只做了解学习，教程中并未配置该环境，因为这边直接用的真机调试**

## 2、安装运行

- 通过命令apt进行安装gazebo

```
sudo apt install ros-${ROS_DISTRO}-gazebo-*
```

- 运行gazebo
- 通过以下命令启动或直接通过桌面图标启动

```
gazebo --verbose -s libgazebo_ros_init.so -s libgazebo_ros_factory.so
```



```

yahboom@yahboom-virtual-machine: ~
yahboom@yahboom-virtual-machine:~$ gazebo --verbose -s libgazebo_ros_init.so -s libgazebo_ros_factory.so
Gazebo multi-robot simulator, version 11.10.2
Copyright (C) 2012 Open Source Robotics Foundation.
Released under the Apache 2 License.
http://gazebosim.org

[Msg] Waiting for master.
Gazebo multi-robot simulator, version 11.10.2
Copyright (C) 2012 Open Source Robotics Foundation.
Released under the Apache 2 License.
http://gazebosim.org

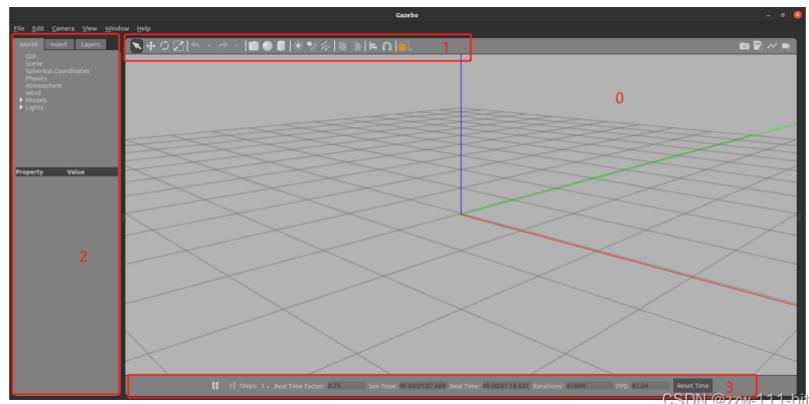
[Wrn] [gazebo_ros_init.cpp:178]
# ##### ###### ####### ###### #####
## # # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# # # # # # # # # #
# ##### ###### ####### #####
This version of Gazebo, now called Gazebo classic, reaches end-of-life
in January 2025. Users are highly encouraged to migrate to the new Gazebo
using our migration guides (https://gazebosim.org/docs/latest/gazebo\_classic\_migration?utm\_source=gazebo\_ros\_pkgs&utm\_medium=cli)

[Msg] Waiting for master.
[Msg] Connected to gazebo master @ http://127.0.0.1:11345
[Msg] Publicized address: 192.168.2.160
[Msg] Loading world file [/usr/share/gazebo-11/worlds/empty.world]
[Msg] Connected to gazebo master @ http://127.0.0.1:11345
[Msg] Publicized address: 192.168.2.160
[Msg] Warning: Ignoring XDG_SESSION_TYPE=wayland on Gnome. Use QT_QPA_PLATFORM=wayland to run on Wayland anyway.
[Wrn] [Event.cc:61] Warning: Deleting a connection right after creation. Make sure to save the ConnectionPtr from a Connect call
context mismatch in svga_surface_destroy

```

运行之后可以看到如下页面：

- 0 : 3D视图区
- 1 : 工具栏
- 2 : 模型列表
- 3 : 模型属性项
- 4 : 时间显示区



- 可选项：为保证模型顺利加载，可以将请将离线模型下载并放置到~/gazebo/models路径下，下载链接如下：[https://github.com/osrf/gazebo\\_models](https://github.com/osrf/gazebo_models)

### 3、gazebo启动节点与服务

#### 1、查看节点

```
ros2 node list
```

```

yahboom@yahboom-virtual-machine: ~ 122x18
yahboom@yahboom-virtual-machine:~$ ros2 node list
/gazebo
yahboom@yahboom-virtual-machine:~$
```

正确返回：/gazebo

#### 2、查看节点的对外提供的服务：

```
ros2 service list
```

可以看出如下的结果：

```
yahboom@yahboom-virtual-machine:~ 122x18
[yahboom@yahboom-virtual-machine:~]$ ros2 node list
/gazebo
[yahboom@yahboom-virtual-machine:~]$ ros2 service list
/delete_entity
/gazebo/describe_parameters
/gazebo/get_parameter_types
/gazebo/get_parameters
/gazebo/list_parameters
/gazebo/set_parameters
/gazebo/set_parameters_atomically
/get_model_list
/pause_physics
/reset_simulation
/reset_world
/spawn_entity
/unpause_physics
[yahboom@yahboom-virtual-machine:~$
```

出去最后几个常规的服务，我们只注意前三个特殊的服务：

- /spawn\_entity，用于加载模型到gazebo中
- /get\_model\_list，用于获取模型列表
- /delete\_entity，用于删除gazebo中已经加载的模型

## 4、创建功能包

- 创建一个myrobot功能包，用来存放我们的URDF模型文件和启动文件

```
ros2 pkg create myrobot --build-type ament_cmake
```

- 进入到myrobot的目录下，创建launch、urdf文件夹，在urdf文件夹下创建一个demo01\_base.urdf文件，这个文件就是一个简单的演示文件，只有一个基础的立方体。

```
<robot name="myrobot">
  <link name="base_link">
    <visual>
      <geometry>
        <box size="0.2 0.2 0.2"/>
      </geometry>
      <origin xyz="0.0 0.0 0.0"/>
    </visual>
    <collision>
      <geometry>
        <box size="0.2 0.2 0.2"/>
      </geometry>
      <origin xyz="0.0 0.0 0.0"/>
    </collision>
    <inertial>
      <mass value="0.1"/>
      <inertia ixx="0.000190416666667" ixy="0" ixz="0" iyy="0.0001904"
      iyz="0" izz="0.00036"/>
    </inertial>
  </link>
  <gazebo reference="base_link">
    <material>Gazebo/Red</material>
  </gazebo>
</robot>
```

## 5、编写launch启动文件

launch文件的编写，launch文件主要启动两个部分，启动Gazebo文件，然后将机器人模型加载到Gazebo中。

```
start_gazebo_cmd = ExecuteProcess(  
    cmd=['gazebo', '--verbose', '-s', 'libgazebo_ros_init.so', '-s',  
    'libgazebo_ros_factory.so'],  
    output='screen')
```

这个命令就是启动Gazebo的，就是一个启动命令，并没有特别复杂的地方，下面是加载模型的命令：

```
spawn_entity_cmd = Node(  
    package='gazebo_ros',  
    executable='spawn_entity.py',  
    arguments=['-entity', robot_name_in_model, '-file', urdf_model_path],  
    output='screen')
```

这个命令注意后面两个参数-entity是模型文件中的名字，-file是通过urdf文件加载参数，后面我们还可以看到通过topic话题加载模型的。在launch目录下新建一个bringup\_model.launch.py文件完整的启动文件如下：

```
import os  
from launch import LaunchDescription  
from launch.actions import ExecuteProcess  
from launch_ros.actions import Node  
from launch_ros.substitutions import FindPackageShare  
from launch_ros.parameter_descriptions import ParameterValue  
from launch.substitutions import Command  
  
def generate_launch_description():  
    robot_name_in_model = 'myrobot'  
    package_name = 'myrobot'  
    urdf_name = "demo01_base.urdf"  
  
    ld = LaunchDescription()  
    pkg_share = FindPackageShare(package=package_name).find(package_name)  
    urdf_model_path = os.path.join(pkg_share, f'urdf/{urdf_name}')  
  
    # Start Gazebo server  
    start_gazebo_cmd = ExecuteProcess(  
        cmd=['gazebo', '--verbose', '-s', 'libgazebo_ros_init.so', '-s',  
        'libgazebo_ros_factory.so'],  
        output='screen')  
  
    # Launch the robot  
    spawn_entity_cmd = Node(  
        package='gazebo_ros',  
        executable='spawn_entity.py',  
        arguments=['-entity', robot_name_in_model, '-file', urdf_model_path],  
        output='screen')  
  
    ld.add_action(start_gazebo_cmd)
```

```
ld.add_action(spawn_entity_cmd)

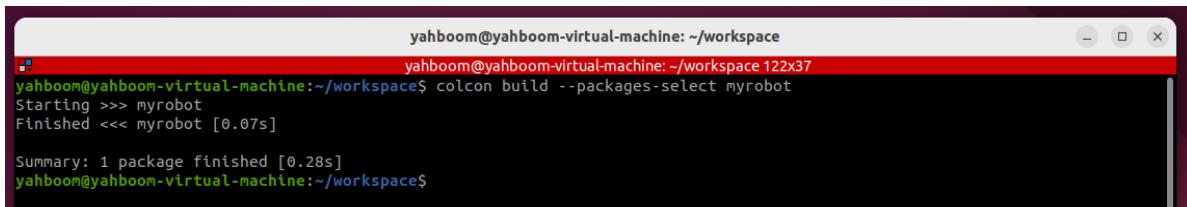
return ld
```

- 在Cmakelist中填入以下内容，用于将我们的urdf和launch文件夹安装进install目录下

```
install(
    DIRECTORY urdf launch
    DESTINATION share/${PROJECT_NAME}
)
```

- 之后编译运行功能包

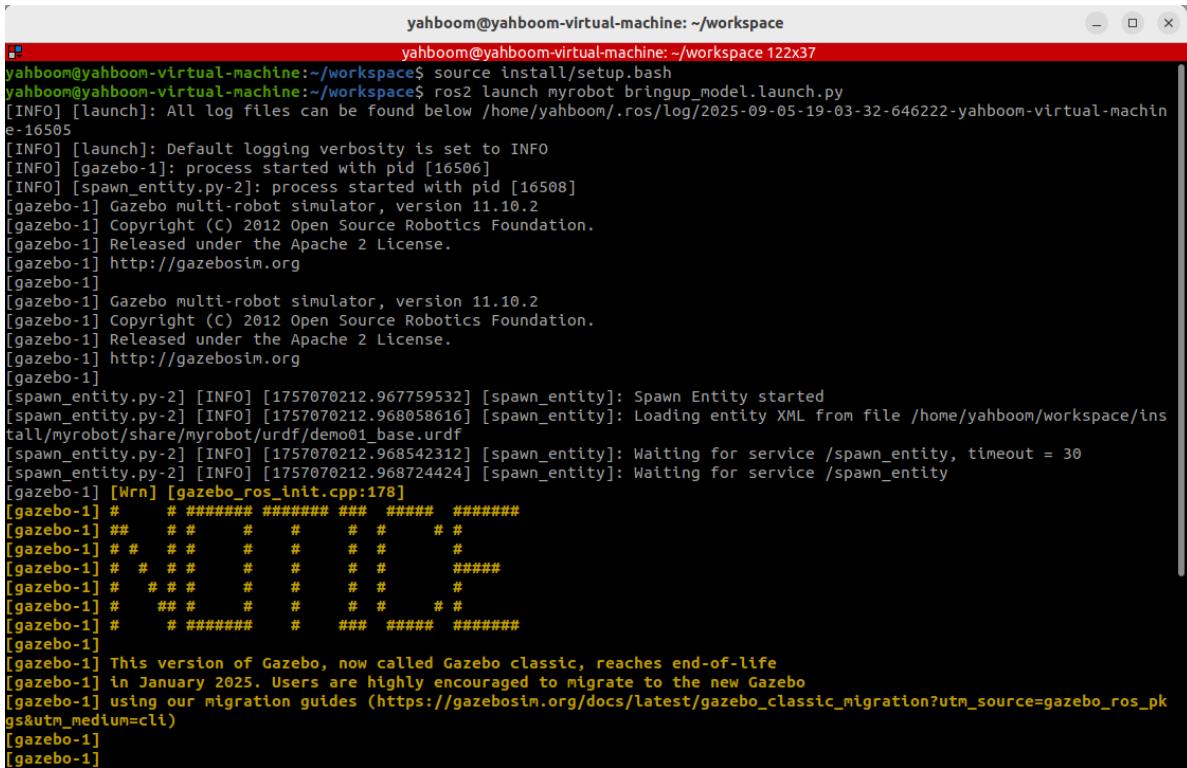
```
colcon build --packages-select myrobot
```



A terminal window titled 'yahboom@yahboom-virtual-machine: ~/workspace'. The command 'colcon build --packages-select myrobot' is run. The output shows the process starting, finishing quickly, and summarizing one package finished in 0.28s. The terminal prompt ends with '\$'.

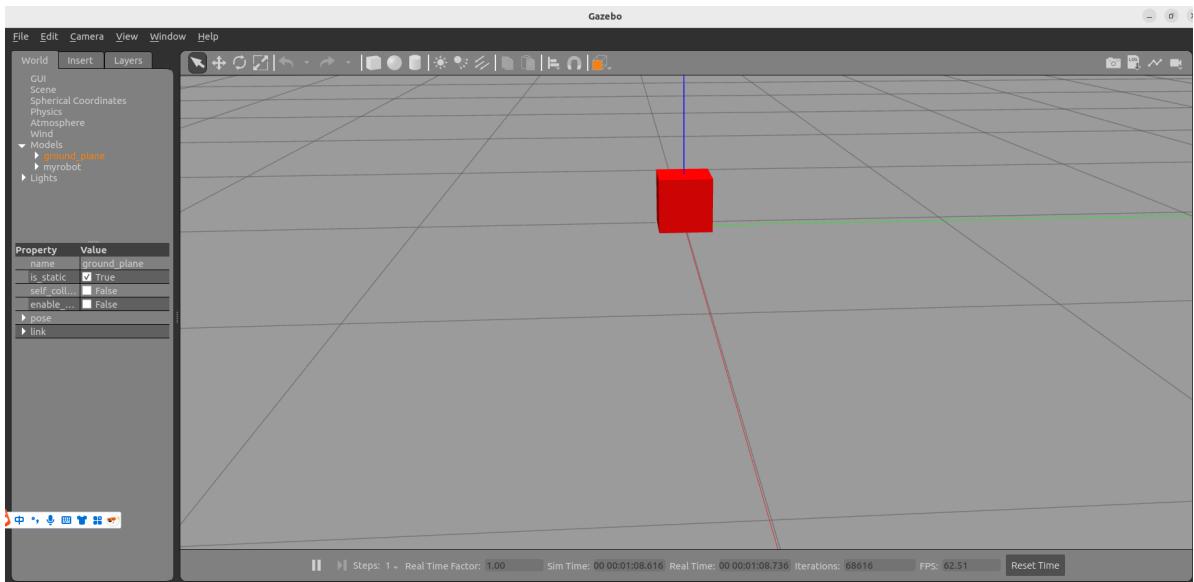
- 刷新环境变量然后运行launch启动文件

```
ros2 launch myrobot bringup_model.launch.py
```



A terminal window titled 'yahboom@yahboom-virtual-machine: ~/workspace'. The command 'source install/setup.bash' is run first. Then 'ros2 launch myrobot bringup\_model.launch.py' is run. The output shows Gazebo starting up, loading the 'base.urdf' file, and waiting for services. It also includes a warning about the end-of-life of Gazebo classic. The terminal prompt ends with '\$'.

启动之后可以看到如下的Gazebo模型：



可以看到红色的模型，因为最后加上了Gazebo的标签设置。