Robot Simulation in Gazebo Using ROS 2

Workspace & File Structure

Ensure all the files are included and our Workspace should look like this

```
my_lidar_description/
├── launch/
|    ├── controller.launch
|    ├── controller.yaml
|    ├── display.launch
|    ├── display.launch.py
|    ├── gazebo.launch.py
|    ├── spawn_entity.py
|    ├── spawn.launch.py
|    └── urdf.rviz
├── meshes/
|    ├── base_link.stl
|    ├── left_wheel_1.stl
|    ├── lidar_1.stl
|    └── right_wheel_1.stl
├── urdf/
|    ├── lidar.xacro
|    ├── lidar.urdf
|    ├── lidar.trans
|    ├── lidar.gazebo
|    └── materials.xacro
├── CMakeLists.txt
└── package.xml
```

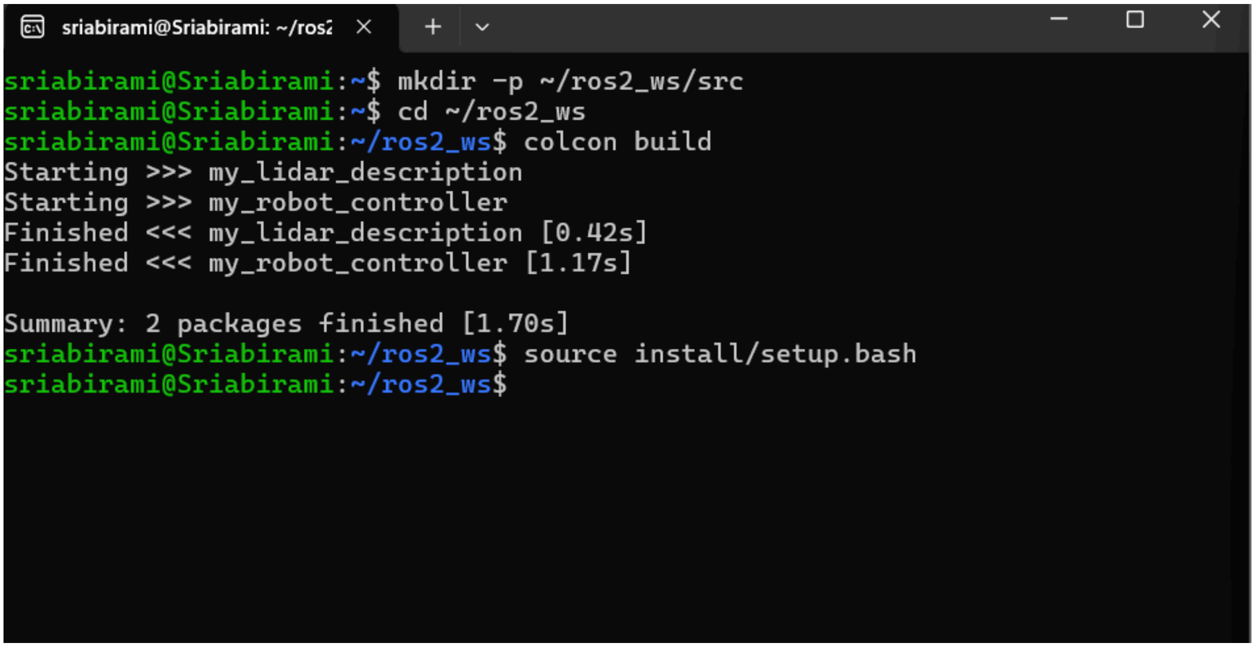
Step-by-Step Instructions

1. Create and Build ROS 2 Workspace

Terminal:

```
mkdir -p ~/ros2_ws/src
cd ~/ros2_ws
```

```
colcon build
source install/setup.bash
```



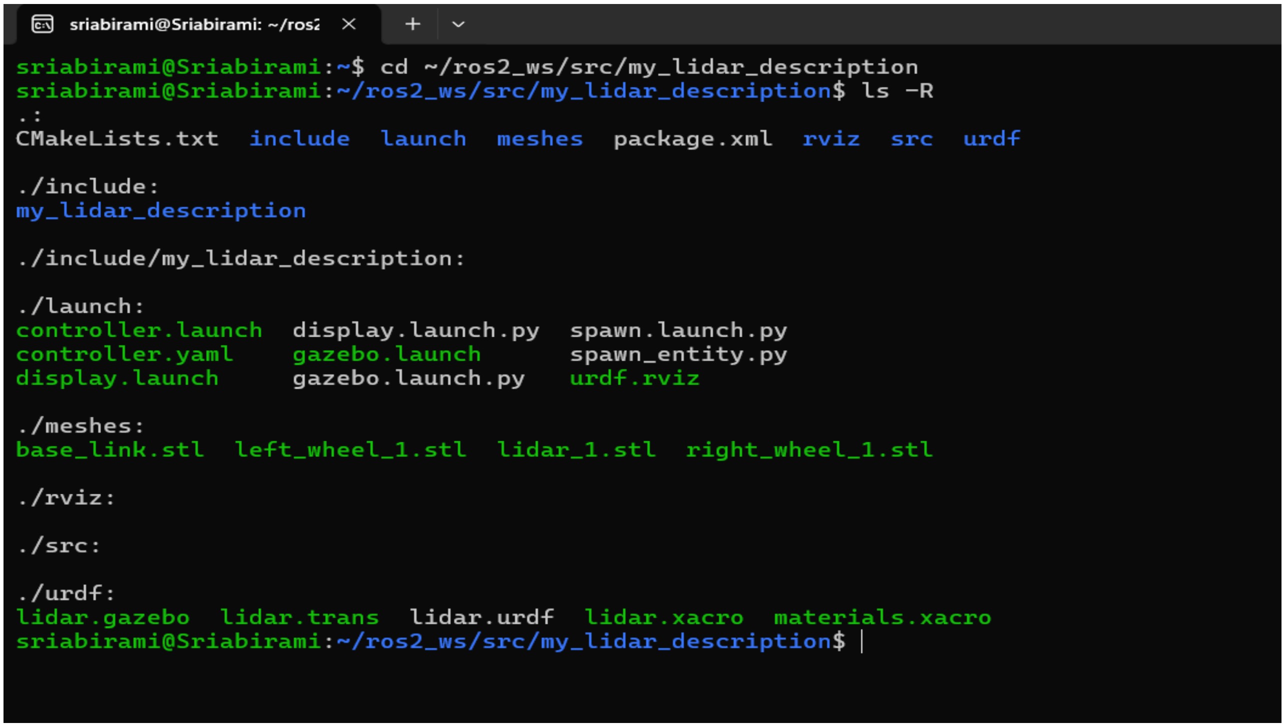
```
sriabirami@Sriabirami:~$ mkdir -p ~/ros2_ws/src
sriabirami@Sriabirami:~$ cd ~/ros2_ws
sriabirami@Sriabirami:~/ros2_ws$ colcon build
Starting >>> my_lidar_description
Starting >>> my_robot_controller
Finished <<< my_lidar_description [0.42s]
Finished <<< my_robot_controller [1.17s]

Summary: 2 packages finished [1.70s]
sriabirami@Sriabirami:~/ros2_ws$ source install/setup.bash
sriabirami@Sriabirami:~/ros2_ws$
```

2. Add Robot Package

Created the folder named **my_lidar_description** and inside **src/** , included our design files.

Terminal:

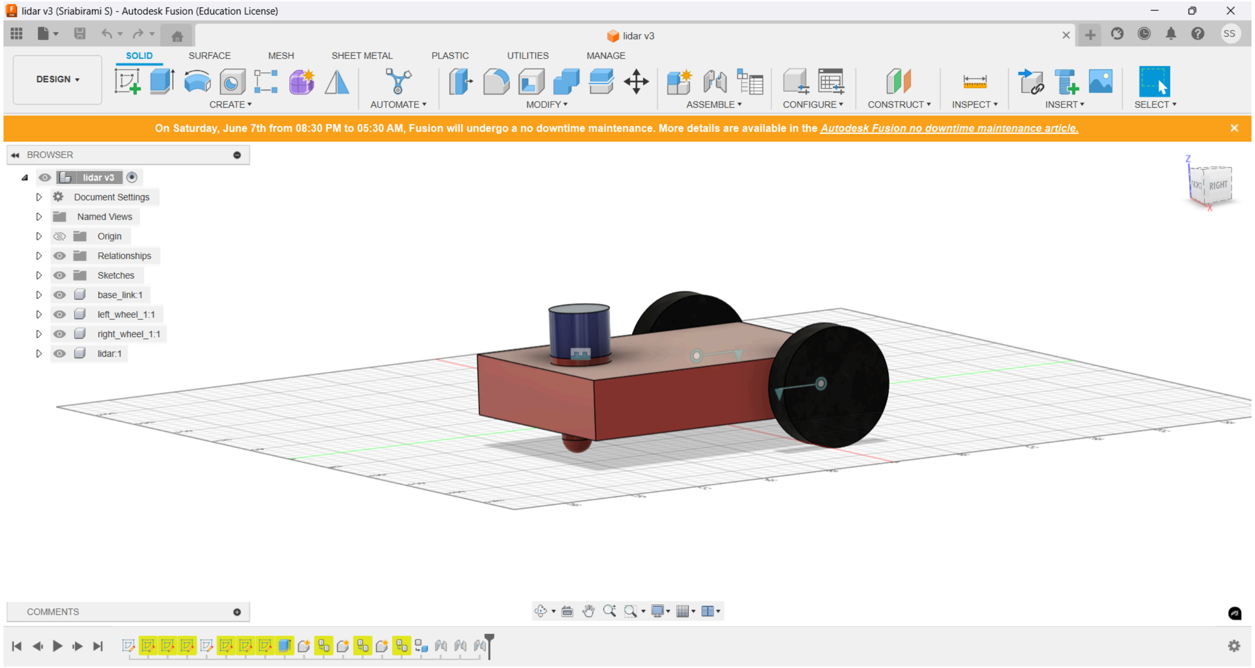
```
cd ~/ros2_ws/src
mkdir -p my_lidar_description/{meshes,urdf,launch}
```



```
sriabirami@Sriabirami:~$ cd ~/ros2_ws/src/my_lidar_description
sriabirami@Sriabirami:~/ros2_ws/src/my_lidar_description$ ls -R
.:
CMakeLists.txt  include  launch  meshes  package.xml  rviz  src  urdf

./include:
my_lidar_description

./include/my_lidar_description:

./launch:
controller.launch  display.launch.py  spawn.launch.py
controller.yaml    gazebo.launch      spawn_entity.py
display.launch     gazebo.launch.py   urdf.rviz

./meshes:
base_link.stl  left_wheel_1.stl  lidar_1.stl  right_wheel_1.stl

./rviz:

./src:

./urdf:
lidar.gazebo  lidar.trans  lidar.urdf  lidar.xacro  materials.xacro
sriabirami@Sriabirami:~/ros2_ws/src/my_lidar_description$ |
```

3. Prepare the Robot Description Package

1. Designed the robot in Fusion 360 and exported it using the urdf exporter Add-in.
2. The entire robot is exported as a URDF package ,such that it automatically creates the necessary folder like meshes/urdf ,launch files etc.
3. Verify all the links,joints and meshes are correctly referenced in the **lidar.xacro** file.
4. Ensure that **`materials.xacro`**, **`lidar.trans`**, and **`lidar.gazebo`** are present and properly referenced in the main xacro file.


lidar v3 (Sriabirami S) - Autodesk Fusion (Education License)

On Saturday, June 7th from 08:30 PM to 05:30 AM, Fusion will undergo a no downtime maintenance. More details are available in the *Autodesk Fusion no downtime maintenance article.*

4. Create Launch File

Create a launch file named **gazebo.launch.py** under launch folder and add the following code:

Code:

```
from launch import LaunchDescription
from launch_ros.actions import Node
from launch.actions import IncludeLaunchDescription
from launch.launch_description_sources import PythonLaunchDescriptionSource
from launch.substitutions import Command, FindExecutable, PathJoinSubstitution
from launch_ros.substitutions import FindPackageShare
```

```
def generate_launch_description():
    pkg_name = 'my_lidar_description'
    pkg_share = FindPackageShare(pkg_name)

    urdf_file = PathJoinSubstitution([pkg_share, 'urdf',
'lidar.xacro'])

    return LaunchDescription([
        Node(
            package='robot_state_publisher',
            executable='robot_state_publisher',
            name='robot_state_publisher',
            output='screen',
            parameters=[{
                'robot_description': Command([
                    FindExecutable(name='xacro'), ' ', urdf_file
                ])
            }]
        ),

        Node(
            package='gazebo_ros',
            executable='spawn_entity.py',
            name='spawn_entity',
            arguments=['-entity', 'lidar_bot', '-topic',
'robot_description'],
            output='screen'
        ),

        IncludeLaunchDescription(
            PythonLaunchDescriptionSource([
                PathJoinSubstitution([FindPackageShare('gazebo_ros'),
'launch', 'gazebo.launch.py'])
            ])
        )
    ])
```

5. Create Custom spawn_entity.py:

Create a file named **spawn_entity.py** in launch folder and add the following code.This code helps to spawn the robot in gazebo

Code:

```
#!/usr/bin/env python3

import rclpy
from gazebo_msgs.srv import SpawnEntity
from ament_index_python.packages import get_package_share_directory
import os

def main():
    rclpy.init()

    node = rclpy.create_node('spawn_entity_client')

    client = node.create_client(SpawnEntity, '/spawn_entity')
    while not client.wait_for_service(timeout_sec=1.0):
        node.get_logger().info('Waiting for /spawn_entity service...')

    sdf_file_path = os.path.join(
        get_package_share_directory('my_lidar_description'),
        'urdf',
        'lidar.xacro'
    )

    with open(sdf_file_path, 'r') as file:
        robot_description = file.read()

    request = SpawnEntity.Request()
    request.name = 'lidar_bot'
    request.xml = robot_description
    request.robot_namespace = ''
    request.reference_frame = 'world'

    future = client.call_async(request)
    rclpy.spin_until_future_complete(node, future)
    if future.result() is not None:
        node.get_logger().info('Successfully spawned entity')
    else:
        node.get_logger().error('Failed to spawn entity')

    rclpy.shutdown()

if __name__ == '__main__':
    main()
```

To make the script executable, run the following in terminal:

Terminal:

```
chmod +x ~/ros2_ws/src/my_lidar_description/scripts/spawn_entity.py
```

6. Build and Source Again

Terminal:

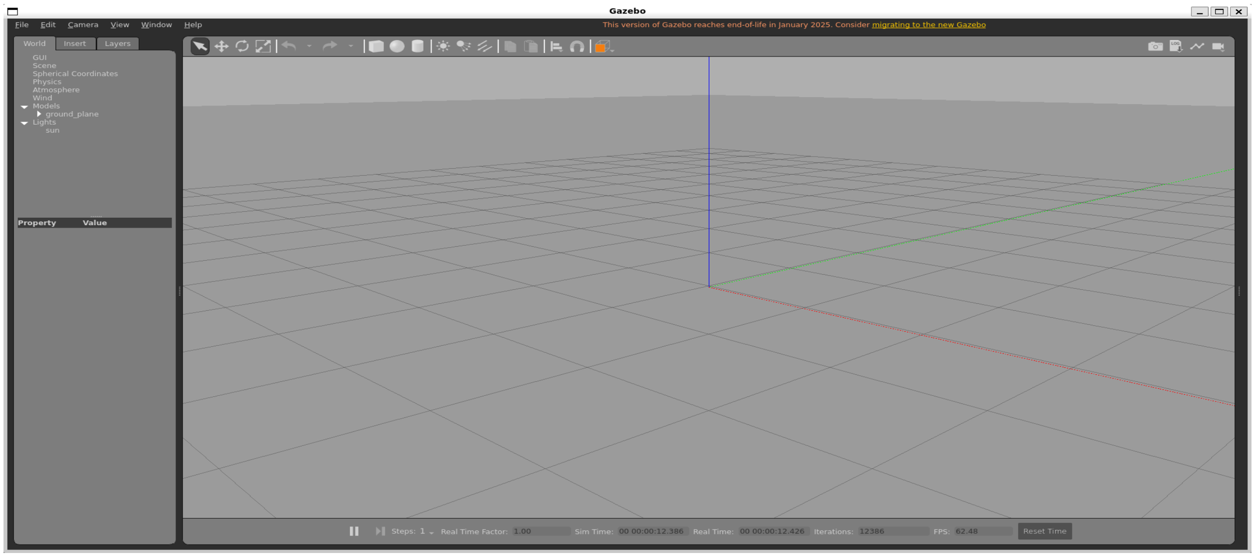
```
cd ~/ros2_ws
colcon build
source install/setup.bash
```

7. Launch Gazebo

First we need to launch the gazebo script to launch the robot in the gazebo

Terminal:

```
ros2 launch gazebo_ros gazebo.launch.py
```



This version of Gazebo reaches end-of-life in January 2025. Consider migrating to the new Gazebo

Steps: 1 Real Time Factor: 1.00 Sim Time: 00 00:00:12.386 Real Time: 00 00:00:12.426 Iterations: 12386 FPS: 62.48

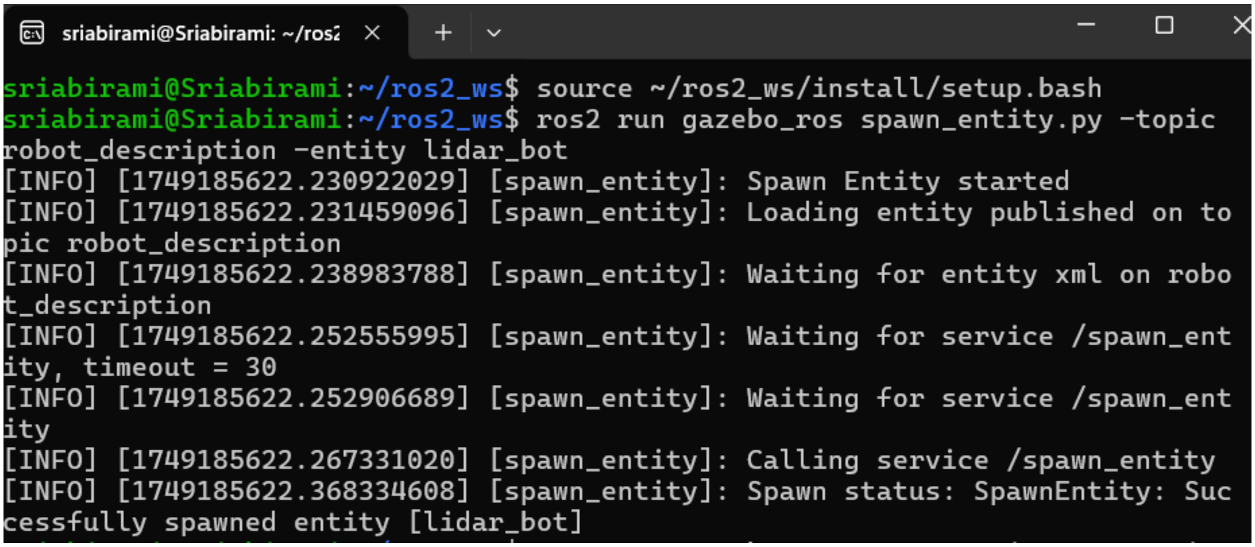
Gazebo Environment Opened

8. Spawn Robot

To make the robot spawn in the gazebo environment run the spawn_entity.py script.

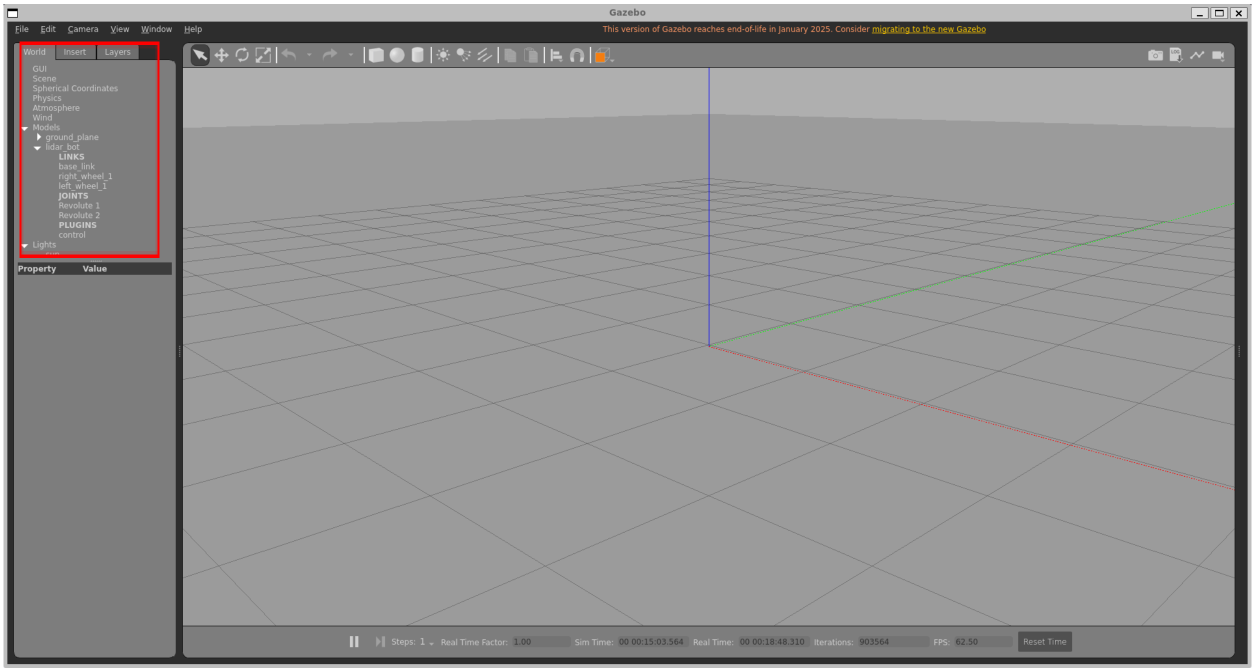
Terminal:

```
ros2 run gazebo_ros spawn_entity.py -topic robot_description -entity
lidar_bot
```



```
sriabirami@Sriabirami:~/ros2_ws$ source ~/ros2_ws/install/setup.bash
sriabirami@Sriabirami:~/ros2_ws$ ros2 run gazebo_ros spawn_entity.py -topic
robot_description -entity lidar_bot
[INFO] [1749185622.230922029] [spawn_entity]: Spawn Entity started
[INFO] [1749185622.231459096] [spawn_entity]: Loading entity published on to
pic robot_description
[INFO] [1749185622.238983788] [spawn_entity]: Waiting for entity xml on robo
t_description
[INFO] [1749185622.252555995] [spawn_entity]: Waiting for service /spawn_ent
ity, timeout = 30
[INFO] [1749185622.252906689] [spawn_entity]: Waiting for service /spawn_ent
ity
[INFO] [1749185622.267331020] [spawn_entity]: Calling service /spawn_entity
[INFO] [1749185622.368334608] [spawn_entity]: Spawn status: SpawnEntity: Suc
cessfully spawned entity [lidar_bot]
```

But the robot was not seen in the gazebo environment and only the models were listed.



This version of Gazebo reaches end-of-life in January 2025. Consider migrating to the new Gazebo

Steps: 1 Real Time Factor: 1.00 Sim Time: 00 00:15:03.564 Real Time: 00 00:18:48.310 Iterations: 903564 FPS: 62.50

Errors faced and Fixes:

1. Entity Already Exists in Gazebo

Error Message:

```
[spawn_entity]: Spawn status: Entity [lidar_bot] already exists.

[ERROR] [spawn_entity]: Spawn service failed. Exiting.
```

Fix: Deleted the existing model from gazebo and launched the robot with a new name

Terminal:

```
ros2 run gazebo_ros spawn_entity.py -topic robot_description -entity
lidar_bot_2 -x 0 -y 0 -z 1
```

2. Robot Not Visible in Gazebo Though Models Are Listed

Fix:

- Ensure the robot is spawned above ground:

Terminal:

```
ros2 run gazebo_ros spawn_entity.py -topic robot_description -entity
lidar_bot -x 0 -y 0 -z 1
```

- Confirm that URDF is actually published:

Terminal:

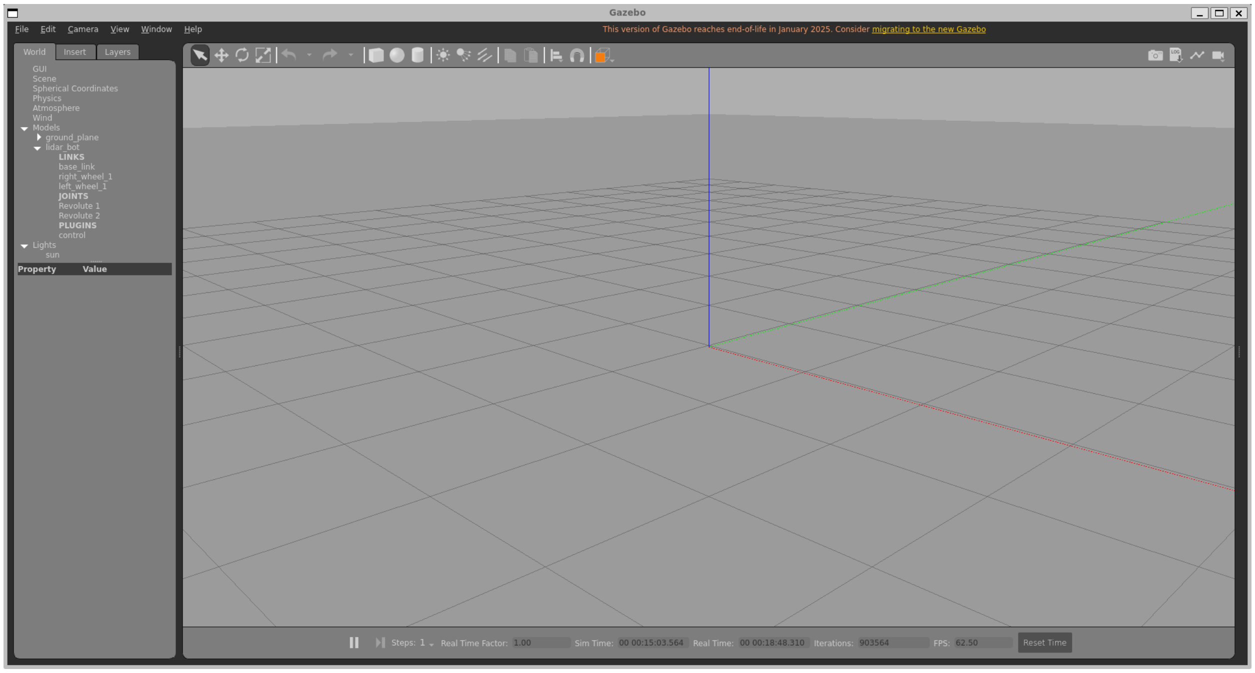
```
ros2 topic echo /robot_description
```

3. Gazebo Window Opens But Robot Doesn’t Spawn Automatically

Fix: Check that the **`gazebo.launch.py`** includes the correct **`spawn_entity`** node. If needed, spawn manually using the command above.

Conclusion:

All the robot parts are visible in the gazebo environment but the design is not visible in the environment.



This version of Gazebo reaches end-of-life in January 2025. Consider migrating to the new Gazebo

Steps: 1 Real Time Factor: 1.00 Sim Time: 00 00:15:03.564 Real Time: 00 00:18:48.310 Iterations: 903564 FPS: 62.50

Video Demonstrations:

Video 1: Opening gazebo simulation

🎬 **ros assingment.mp4**

Video 2: Alternative

🎬 **gazebo simulation of another robot.mp4**

Tried to simulate another robot in gazebo using keyboard control.This video is the simulation video of that robot.