Module 1 Assignment: Student Information Sheet & Setting up a ROS Environment

Introduction to Robotics

Student Survey

1. What programming languages are you proficient in? (e.g. Python2, MATLAB, C++, etc.)

[Dilip:] C++

2. How would you rate your proficiency level in the following subjects?
    a. Linux OS - **[Dilip:]** Medium level
    b. Linear Algebra **[Dilip:]** Medium level
    c. Calculus **[Dilip:]** Medium level
    d. Physics **[Dilip:]** Medium level

3. What are you looking to get out of this course?

[Dilip:] My larger goal is to get hands-on to develop and deploy robots for manufacturing plants. I hope that this course will provide me with the foundation of robotics knowledge which I can apply in future.

4. Which topics listed on the class syllabus are the most interesting to you?

[Dilip:] All the topics listed in the course syllabus are interesting to me. Be it ROS or controls or localization etc, I am excited to explore all these topics to get in-depth knowledge on robotics.

Setting up ROS

Following the instructions in the ROS Setup Handout install ROS2 and the Turtlebot3 simulation on your machine. Then complete the following.

1. Document your system setup (Host OS, VM or Docker)

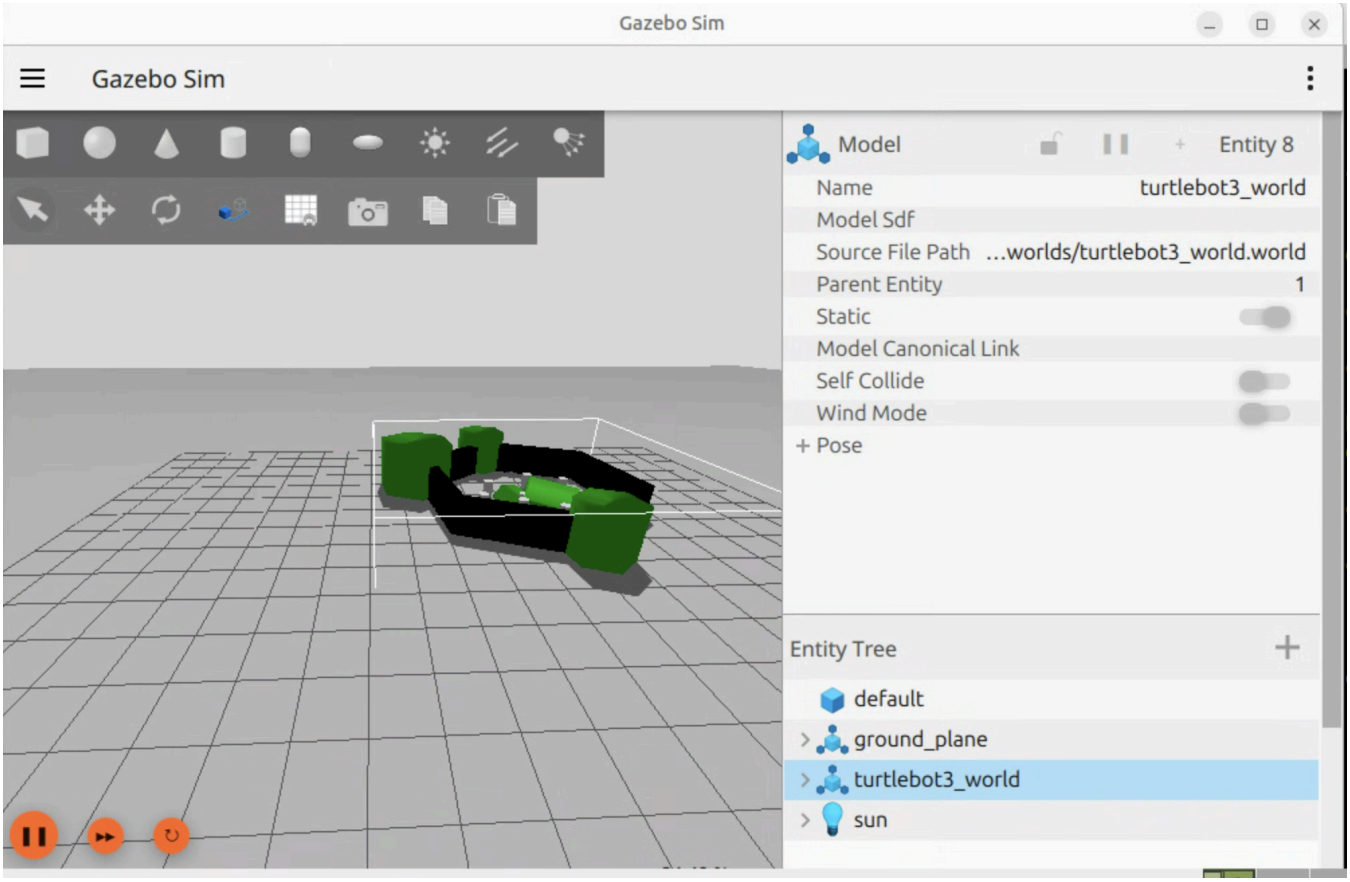
a. Machine type (Desktop, Laptop, Cloud Virtual Machine): **[Dilip:]** Virtual Machine

b. Host Operating system (Windows, Linux, Mac): **[Dilip:]** Linux

c. ROS install type (Local, Virtual Machine, Docker): **[Dilip:]** Docker

2. On a scale of 1-5 how difficult did you find the installation process (1 is easy, 5 is hard)?

[Dilip:] Installation was harder. This was bcz seems like the Waffle model is not compatible with Gazebo Sim (the new, orange-themed simulator). Looks like instruction is based on Gazebo Classic (the older simulator). The Waffle model includes complex sensors (Intel RealSense Depth Camera) and plugins that often have not yet been fully updated or bridged for Gazebo Sim in the early releases of ROS 2 Jazzy. Since the simulator cannot load these specific camera plugins or meshes, the entire robot fails to spawn, resulting in the empty world as below.

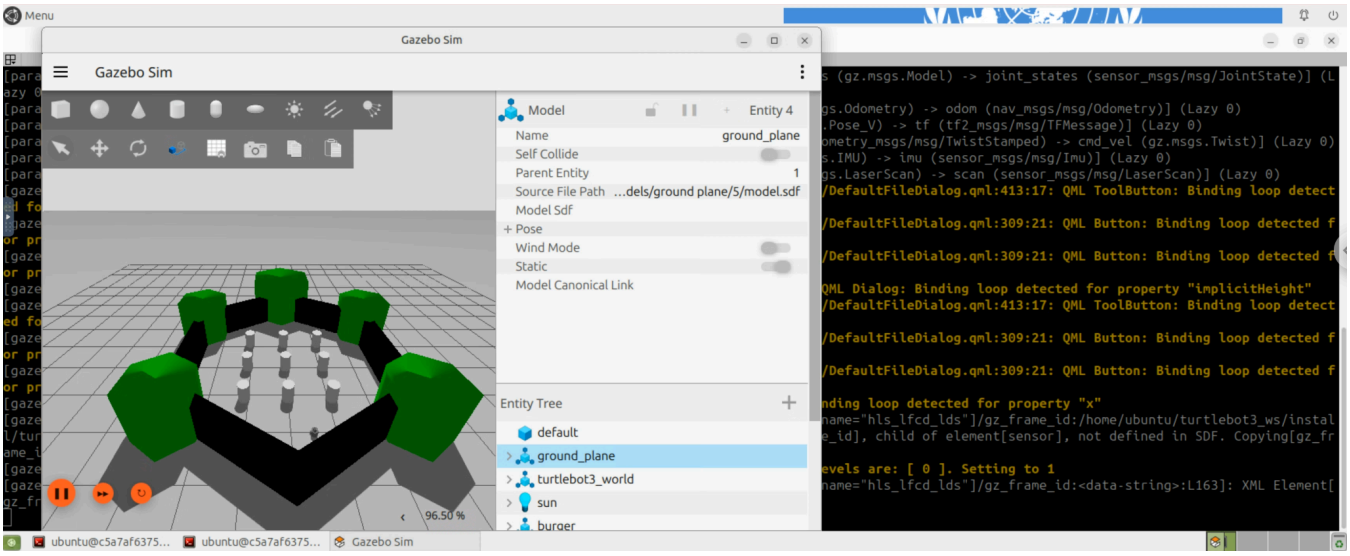


To fix it, I had to mix with the following command.

export TURTLEBOT3_MODEL=burger

ros2 launch turtlebot3_gazebo turtlebot3_world.launch.py

This helped me to launch the simulator with a robot as below.



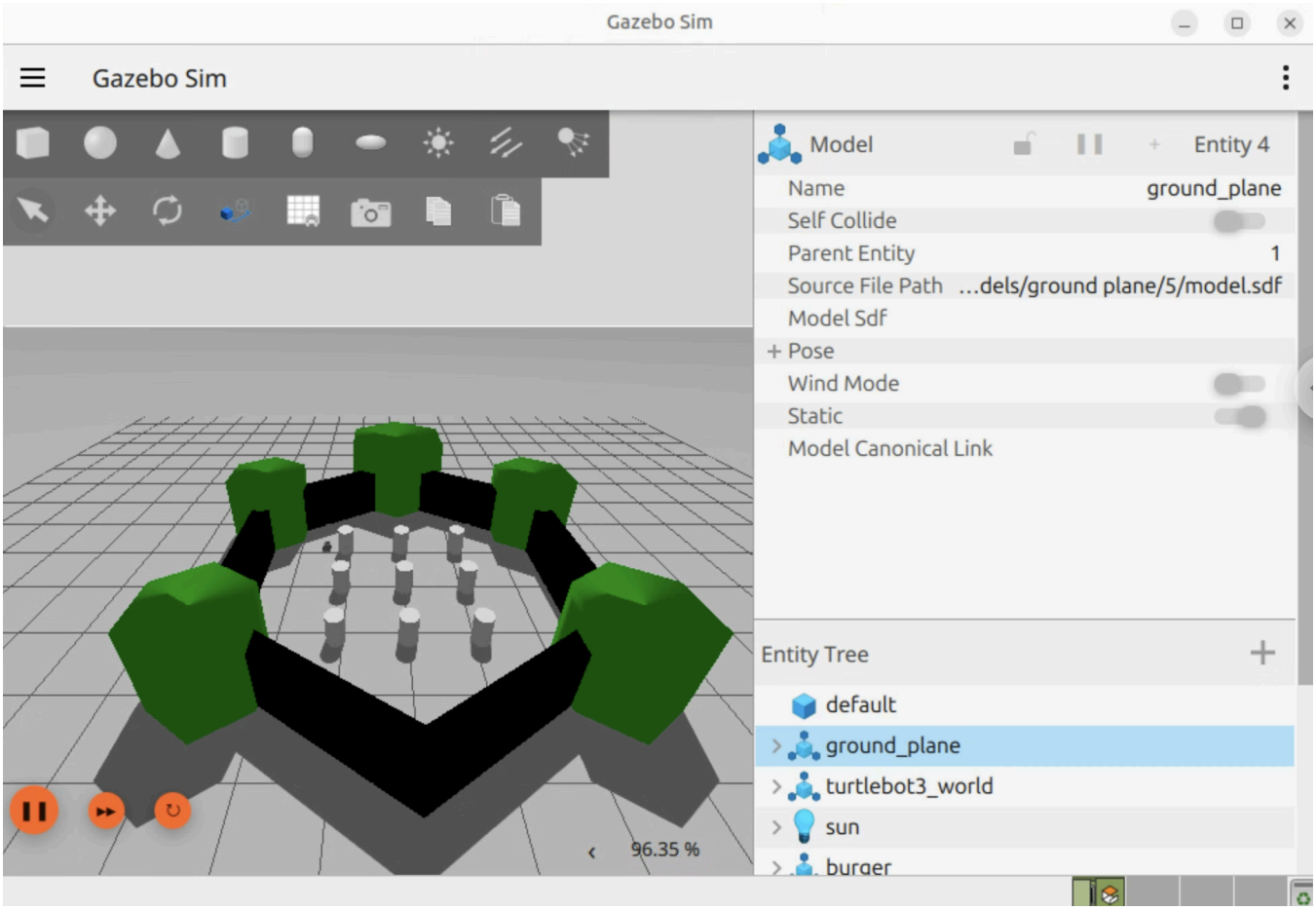
3. Drive the robot to the opposite side of the obstacle field. Paste a screenshot below.

[Dilip:] Ran following command to enable the keyboard control.

export TURTLEBOT3_MODEL=burger

ros2 run turtlebot3_teleop teleop_keyboard

After that, using the combination of linear and angular velocity, moved the robot to the opposite side of the obstacle field (i.e. pillars) as below.


Gazebo Sim
Gazebo Sim
Model
Entity 4
Name
ground_plane
Self Collide
Parent Entity
1
Source File Path
...dels/ground plane/5/model.sdf
Model Sdf
Pose
Wind Mode
Static
Model Canonical Link
Entity Tree
default
ground_plane
turtlebot3_world
sun
burger
96.35 %