Simulation and Modelling Lab – 7

Turtle Bot and Irritated Robot

20BRS1193

Meher Shrishti Nigam

Instructions:

Today Exercise will be on Irritated Robot:

If the robot encounters an obstacle at a threshold distance of 0.5, then the robot engage in a twisted motion or circular motion

If no obstacle, the robot moves forward with a nominal speed.

You can make use of Burger as the Turtlebot model

Exercise 2: From the above example, make the irritated robot to a diplomat robot where the robot moves away from the obstacle and move forward with a nominal speed.

How to do: Open a terminal

Step 1: $ roscore

Step 2: Open another terminal

$ export TURTLEBOT3_MODEL=burger

$ roslaunch turtlebot3_gazebo turtlebot3_empty_world.launch

Step 3:

Copy the irritated_robot.py from the following folder

$ scp lab21@172.16.10.7:irritational_robot.py

move the file to pradeep_ws/src/ros_tutorial1/src/

Go to the above folder and execute the following command.

$ chmod 777 irritational_robot.py

$ rosrun ros_tutorial1 irritational_robot.py

and check the gazebo platform and add the obstacles and check the status of the robot (Burger).

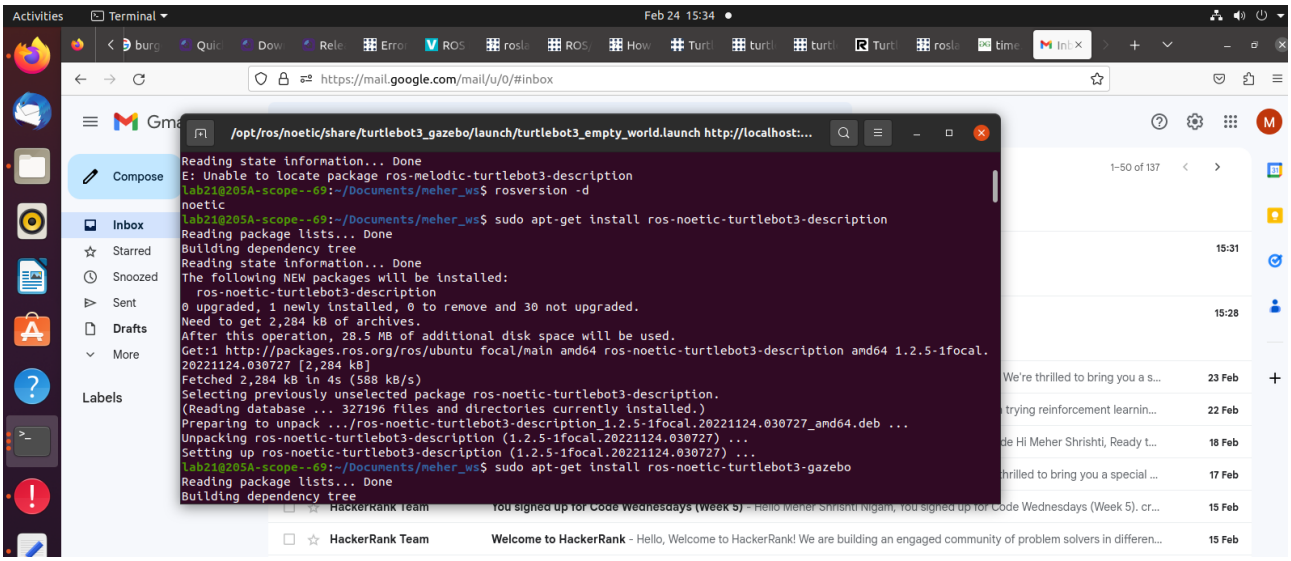
Code:

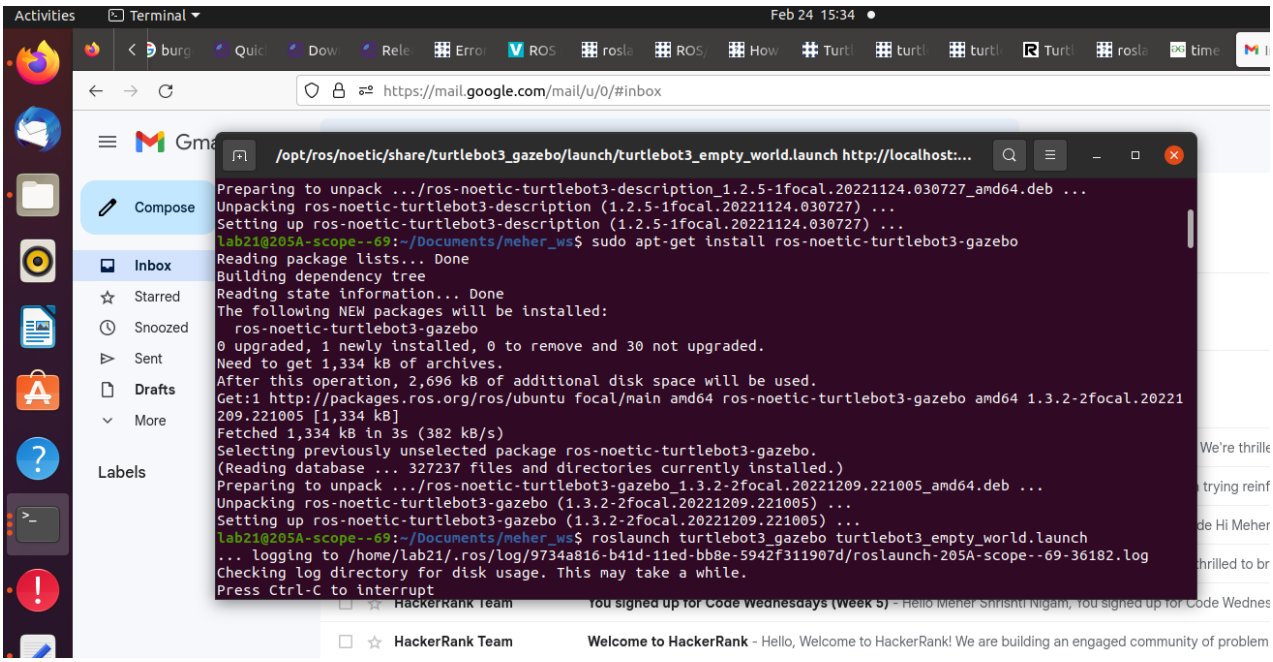
```
#!/usr/bin/python3

import rospy
import numpy as np
from numpy import inf
from geometry_msgs.msg import Twist
from sensor_msgs.msg import LaserScan
import time
import sys

class object_irritation_robot:

    def __init__(self):
        rospy.Subscriber("/scan", LaserScan, self.laserData_cb)
        self.pub = rospy.Publisher('/cmd_vel', Twist, queue_size=10)
        self.robot_velocity=Twist()

    def laserData_cb(self,data):
        laser_data=np.array(data.ranges)
        laser_data[laser_data == inf] = 0
        laser_data=max(laser_data)
        rospy.loginfo(laser_data)

        if(laser_data > 0.5):
            if(self.robot_velocity.linear.x==0.5):
                self.evade()
            else:
                self.move_forward()
        else:
            self.move_forward()

        self.pub.publish(self.robot_velocity)
"""
    def irritated(self):
        rospy.loginfo("I am Irritated")
        self.robot_velocity.linear.x=0.0
        self.robot_velocity.angular.z=2.0
"""
    def move_forward(self):
        rospy.loginfo("Lets Moving ON ")
        self.robot_velocity.linear.x=0.5
```

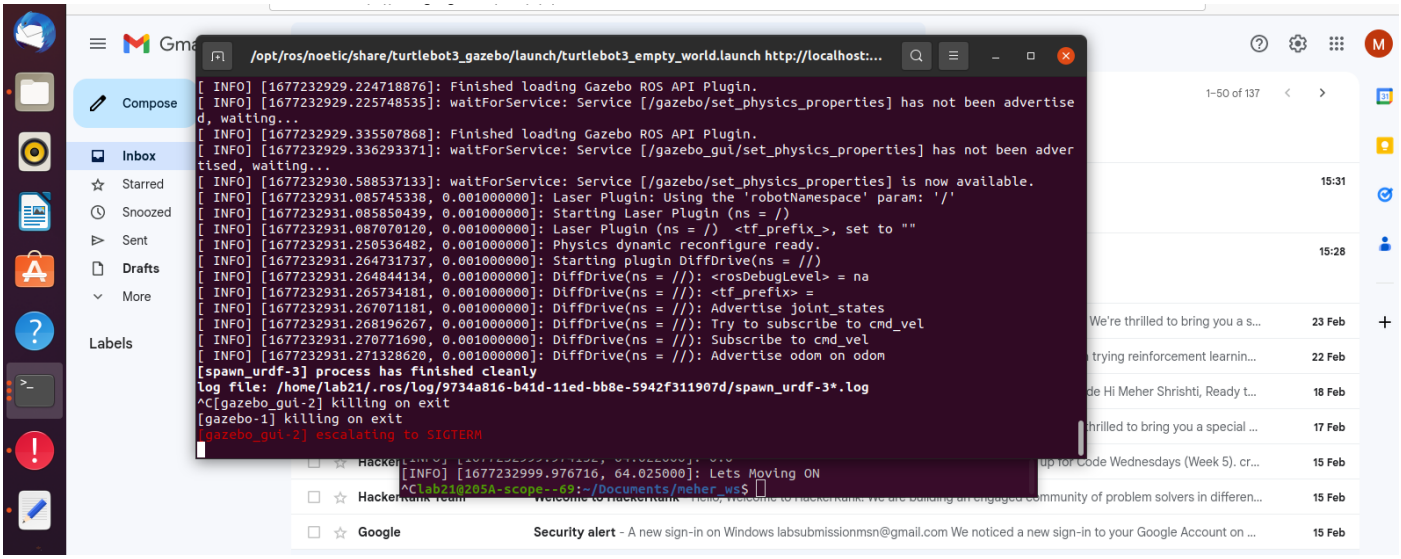
```
        self.robot_velocity.angular.z=0.0

    def evade(self):
        rospy.loginfo("EVADE")
        self.robot_velocity.linear.x=-0.5
        self.robot_velocity.angular.z=3


if __name__ == '__main__':
    rospy.init_node('object_irritation_robot', anonymous=True)
    object_irritation_robot()
    rospy.spin()
```

Terminal Outputs:

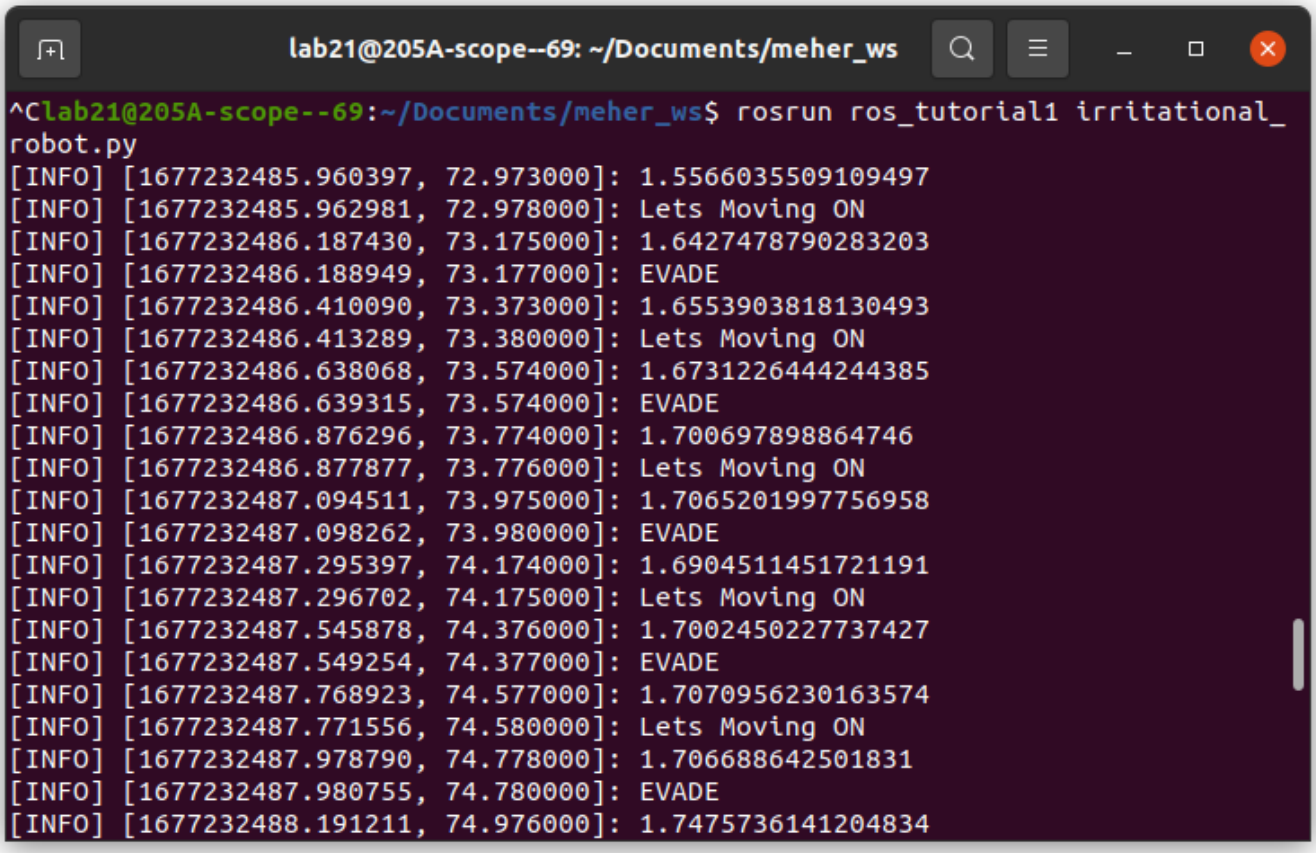
First Install turtlebot3_gazebo, then run the roslaunch command.





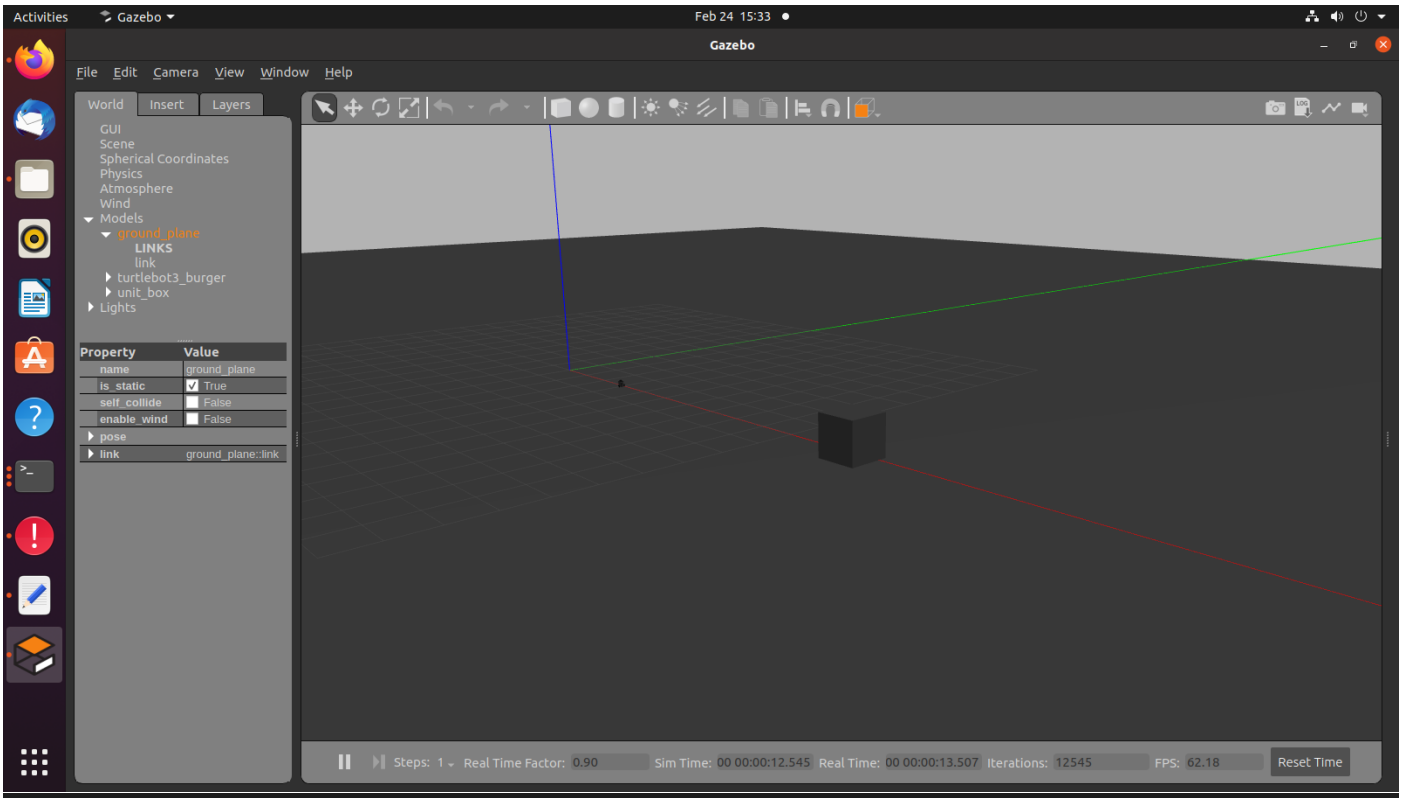


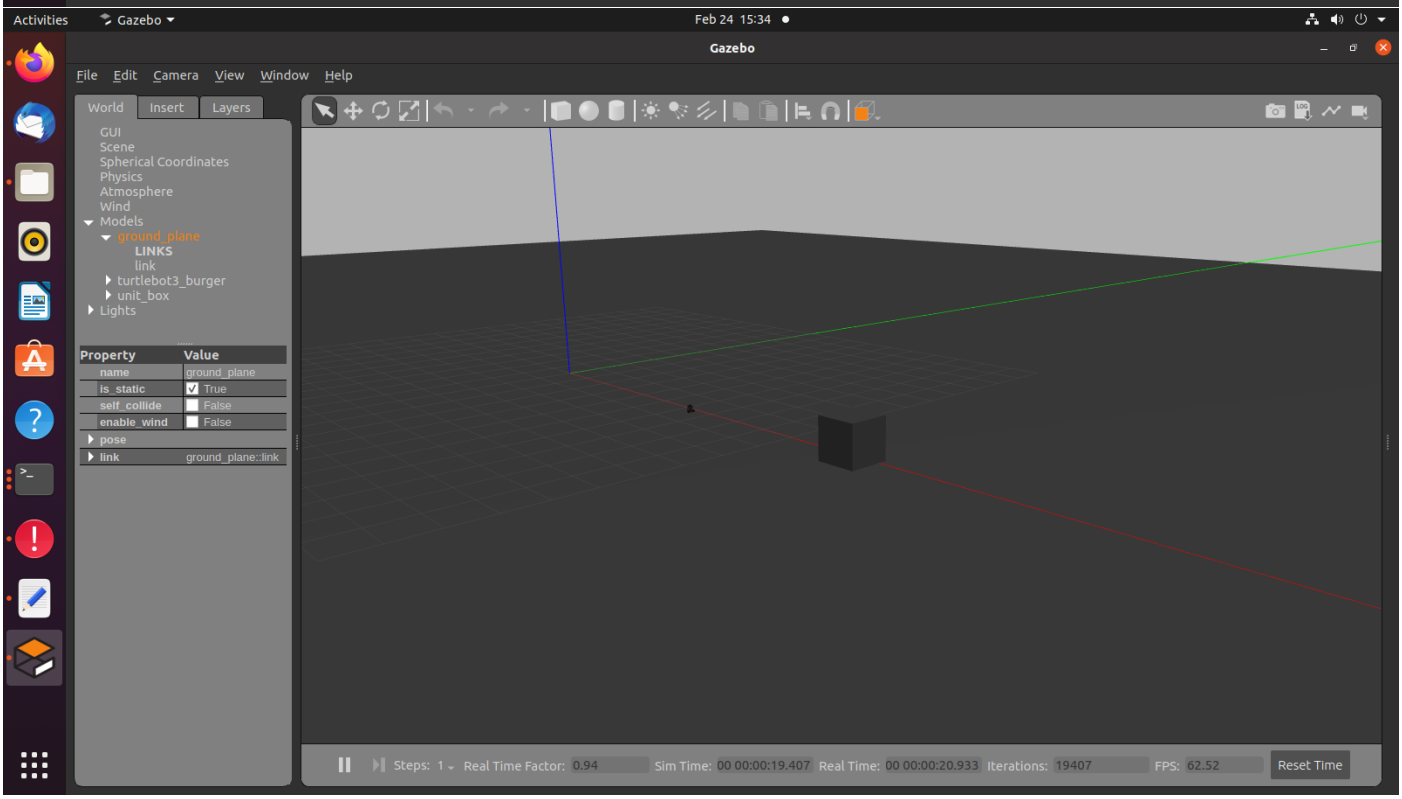
Then run irritational_robot.py using rosrun.

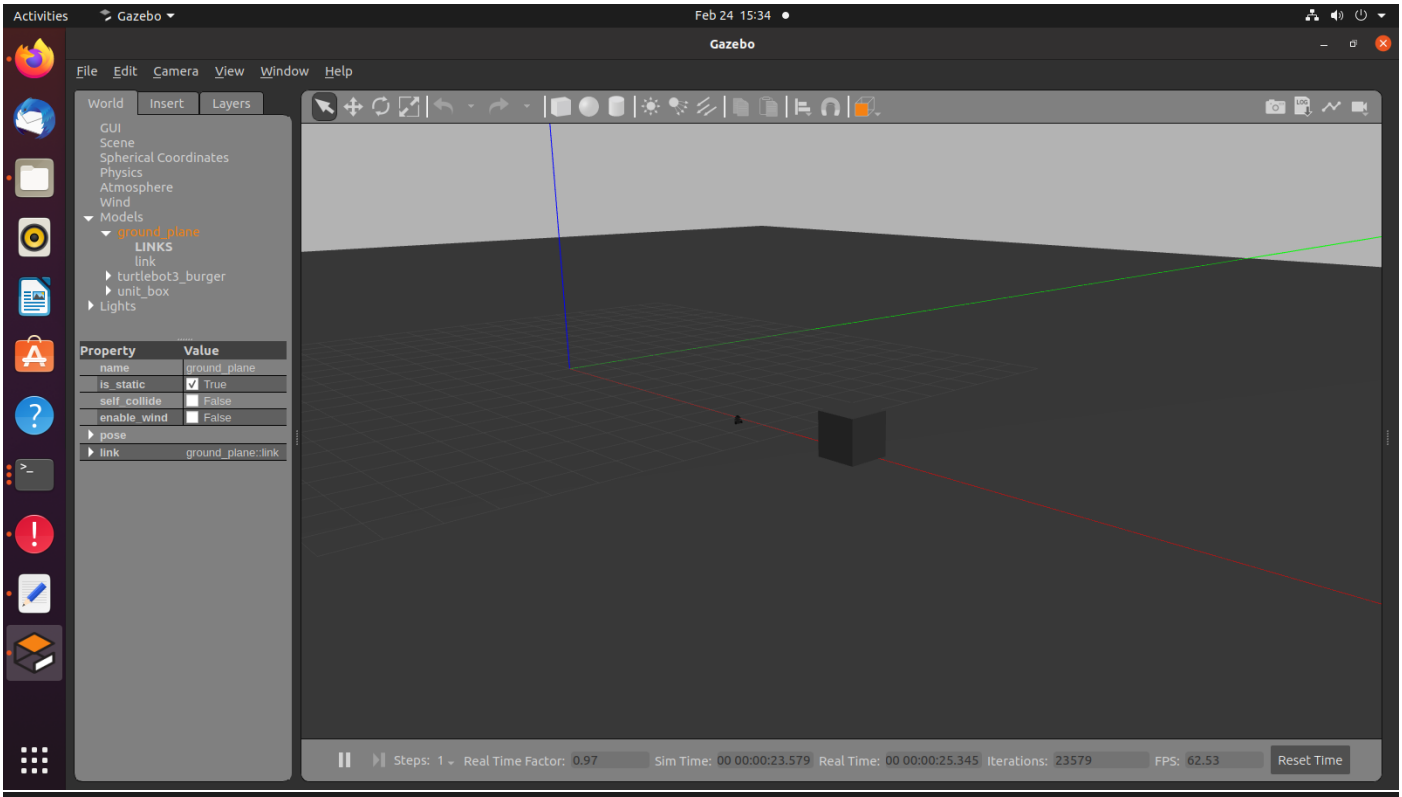


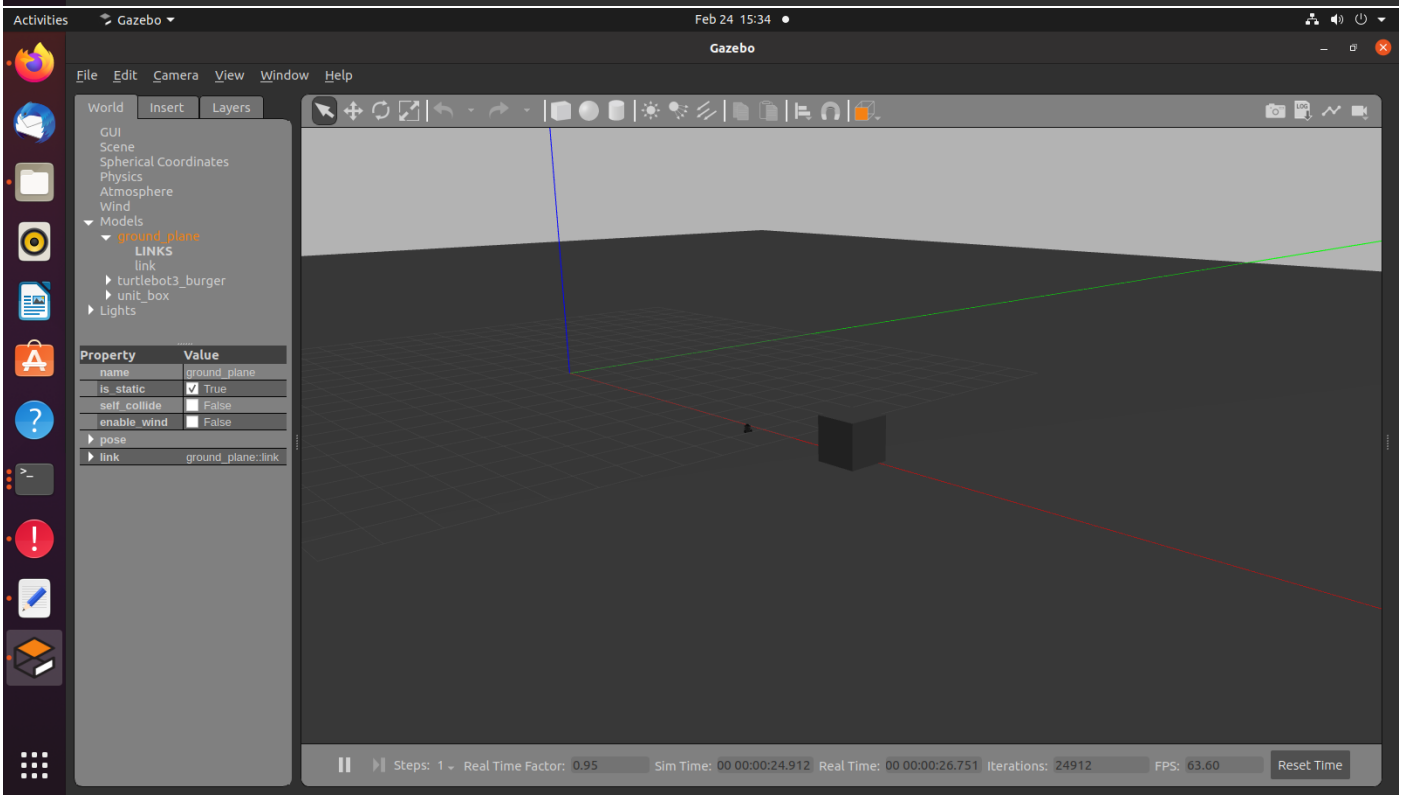
Gazebo Outputs:

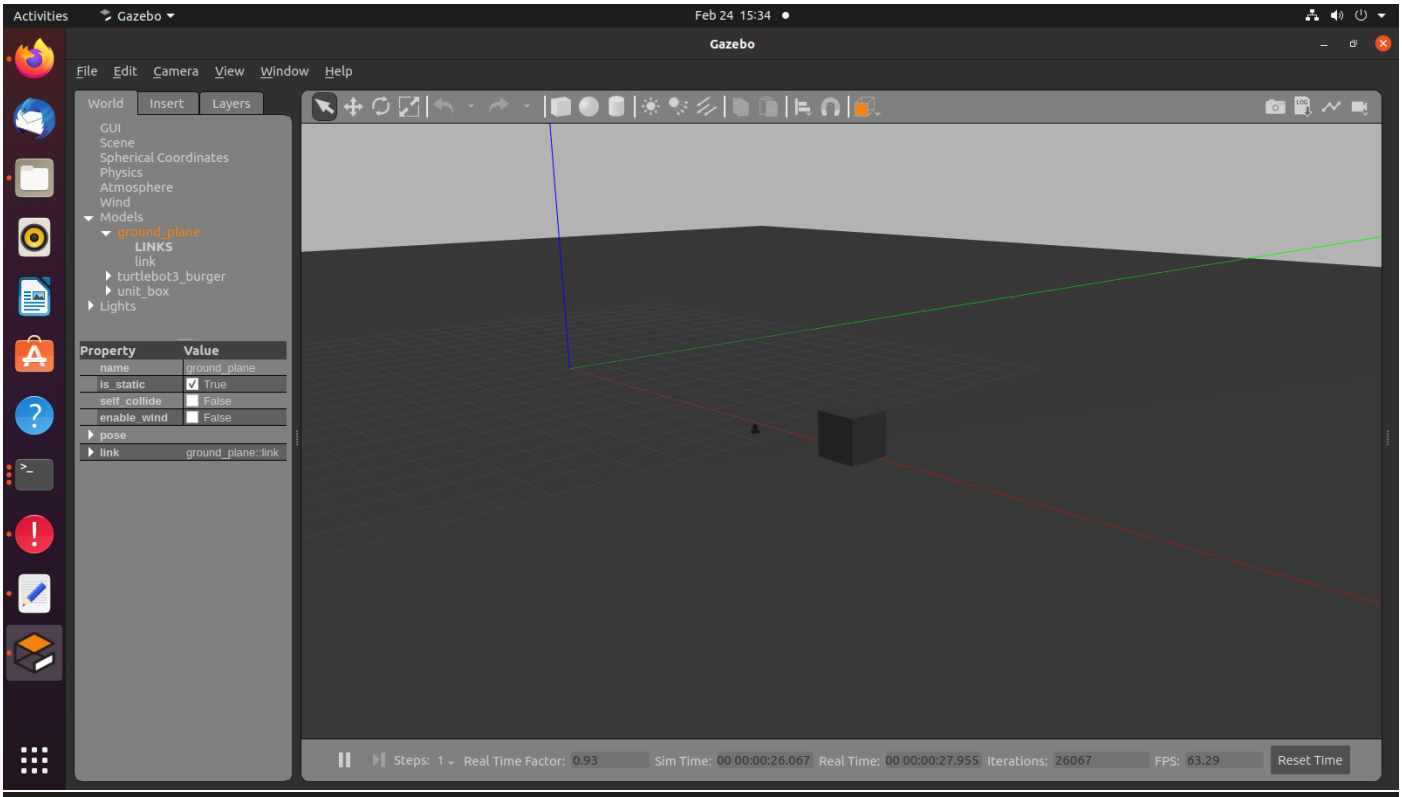
Thus, the robot successfully evades the obstacle.

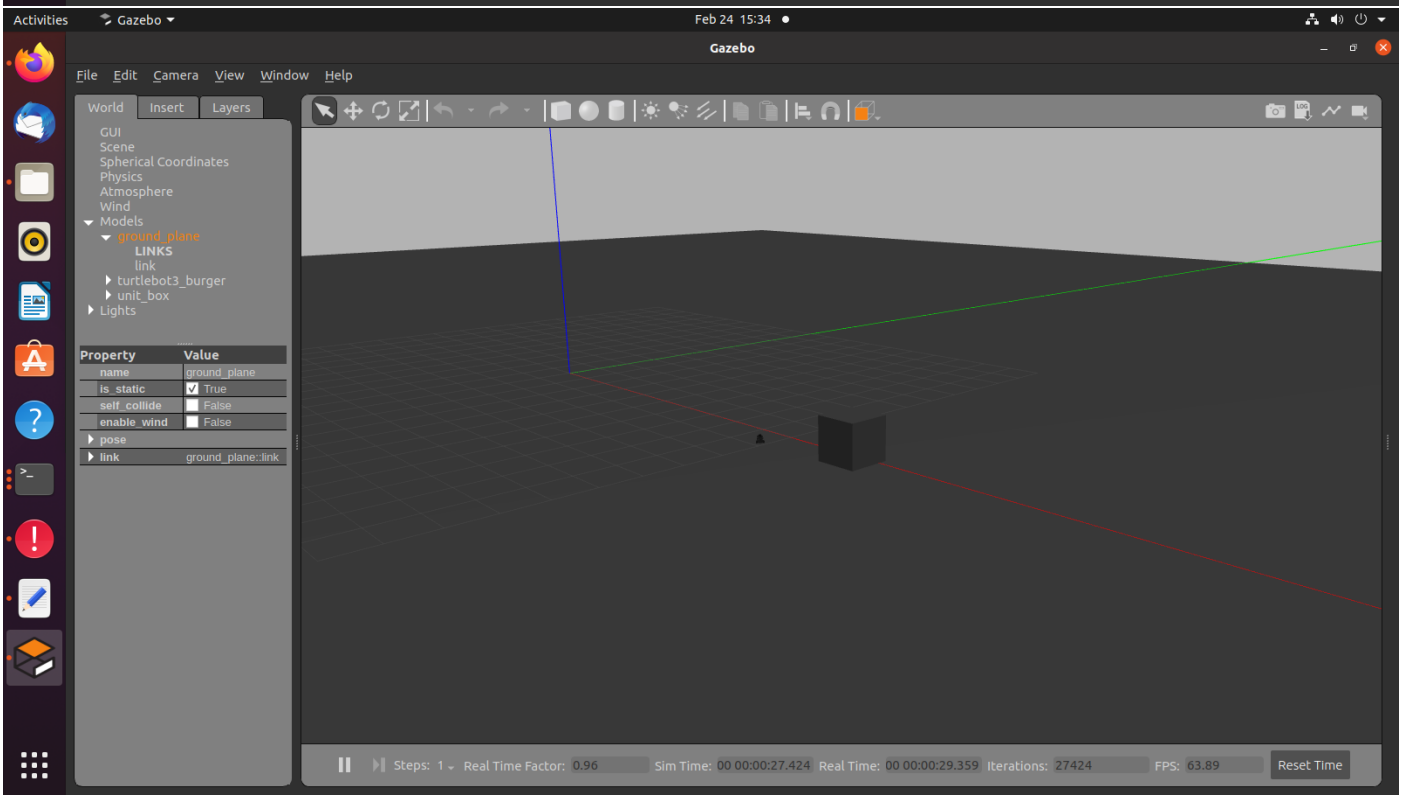


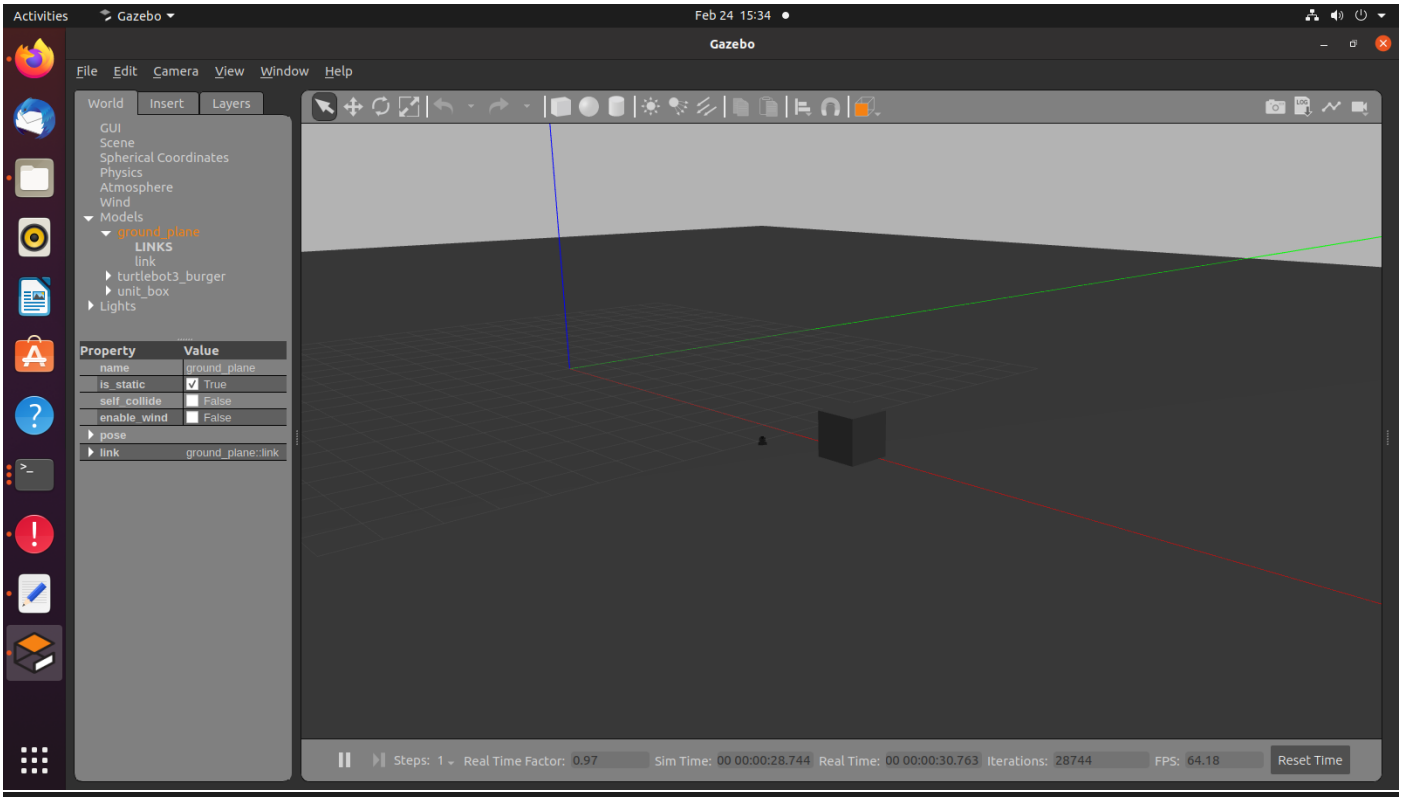


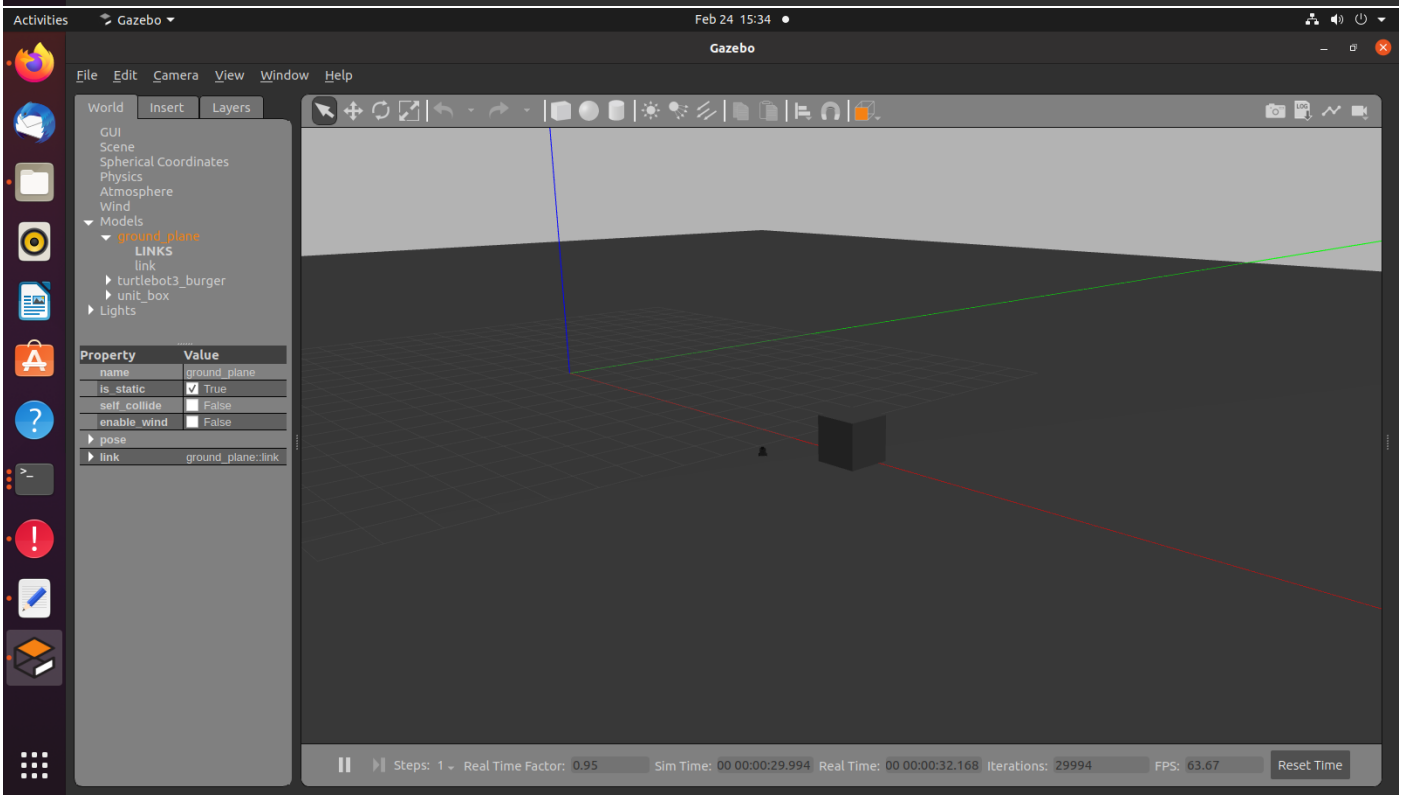

Activities
Gazebo
Feb 24 15:34
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
ground_plane
LINKS
link
turtlebot3_burger
unit_box
Lights
Property Value
name ground_plane
is_static True
self_collide False
enable_wind False
pose
link ground_plane::link
Steps: 1 Real Time Factor: 0.97 Sim Time: 00 00:00:23.579 Real Time: 00 00:00:25.345 Iterations: 23579 FPS: 62.53
Reset Time

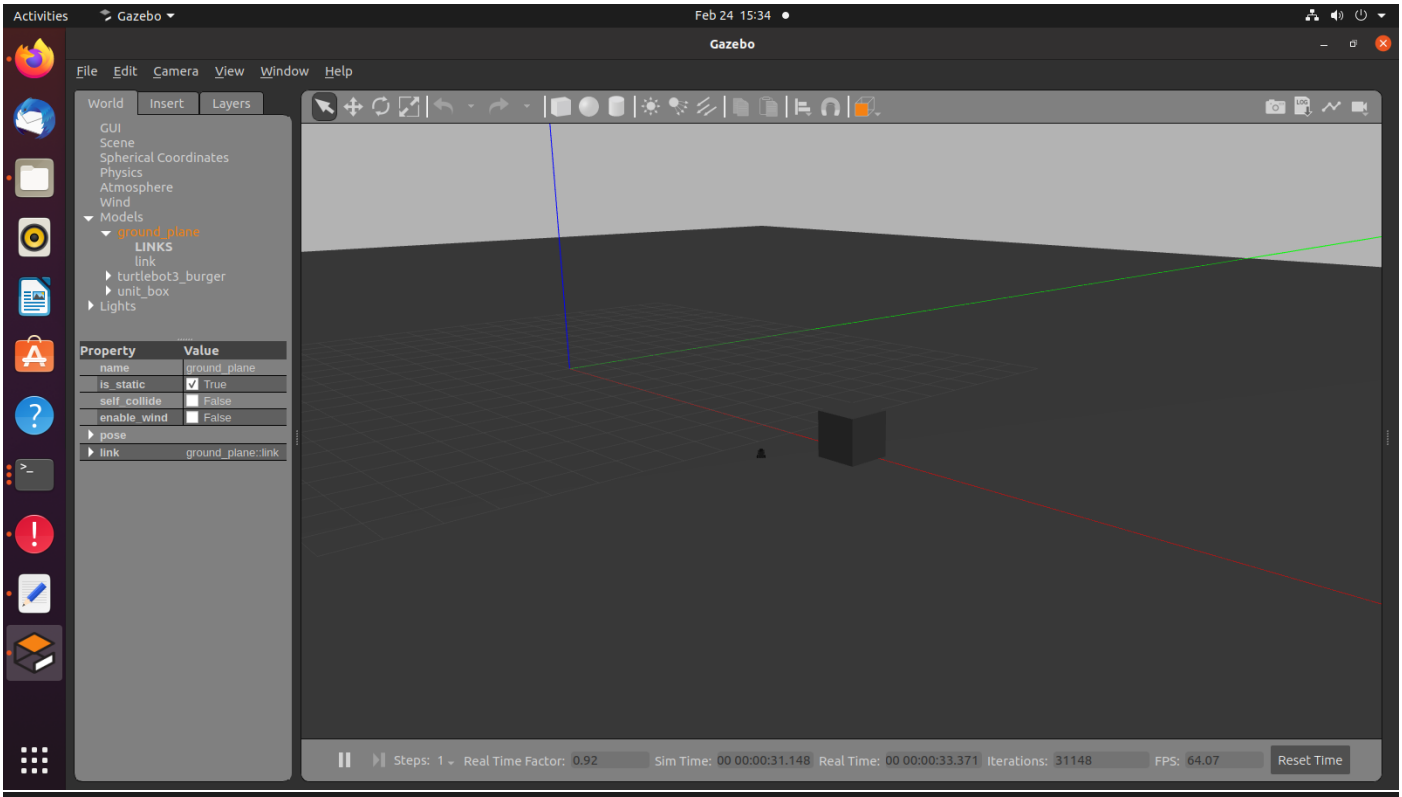

Activities
Gazebo
Feb 24 15:34
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
ground_plane
LINKS
link
turtlebot3_burger
unit_box
Lights
Property Value
name ground_plane
is_static True
self_collide False
enable_wind False
pose
link ground_plane::link
Steps: 1 Real Time Factor: 0.95 Sim Time: 00 00:00:24.912 Real Time: 00 00:00:26.751 Iterations: 24912 FPS: 63.60
Reset Time

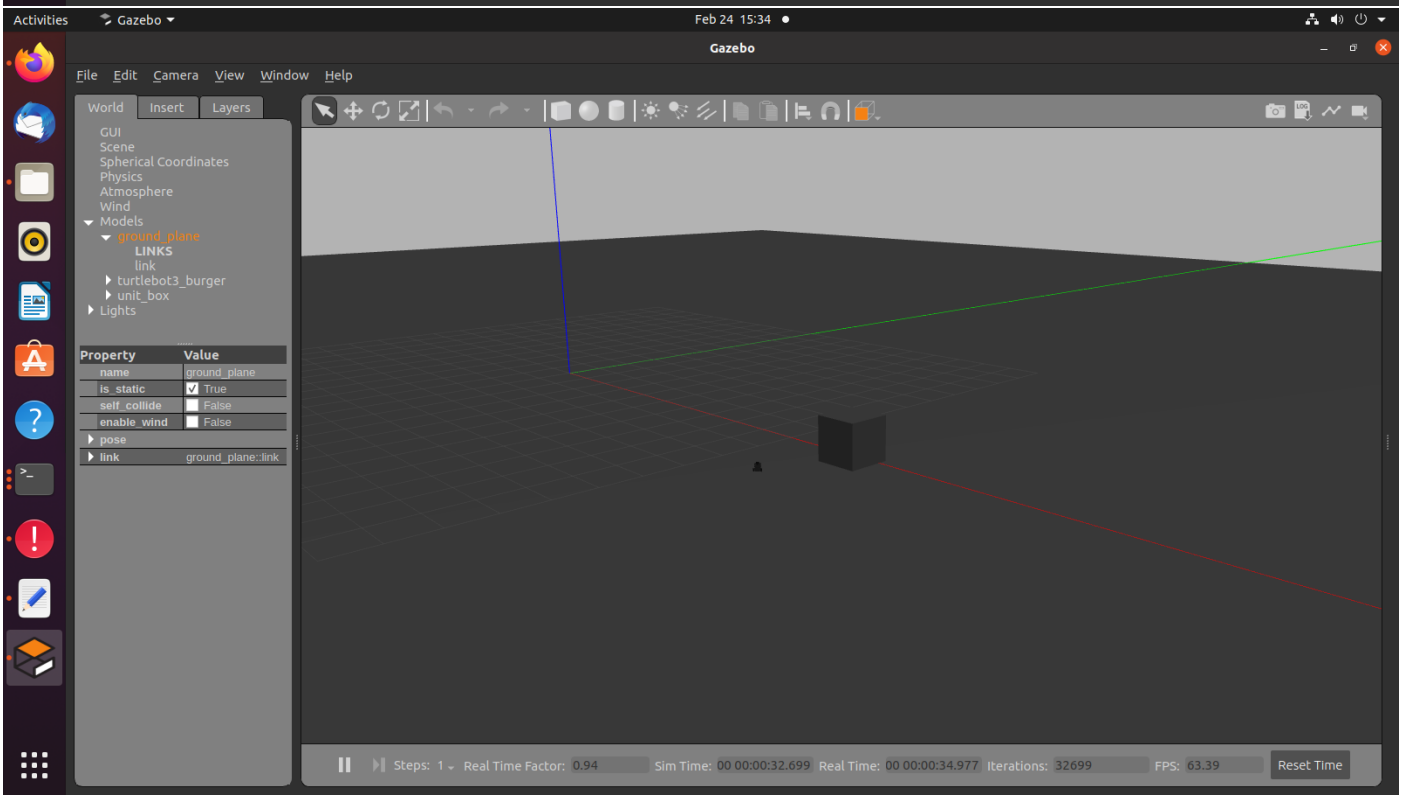

Activities
Gazebo
Feb 24 15:34
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
ground_plane
LINKS
link
turtlebot3_burger
unit_box
Lights
Property Value
name ground_plane
is_static True
self_collide False
enable_wind False
pose
link ground_plane::link
Steps: 1 Real Time Factor: 0.93 Sim Time: 00 00:00:26.067 Real Time: 00 00:00:27.955 Iterations: 26067 FPS: 63.29
Reset Time

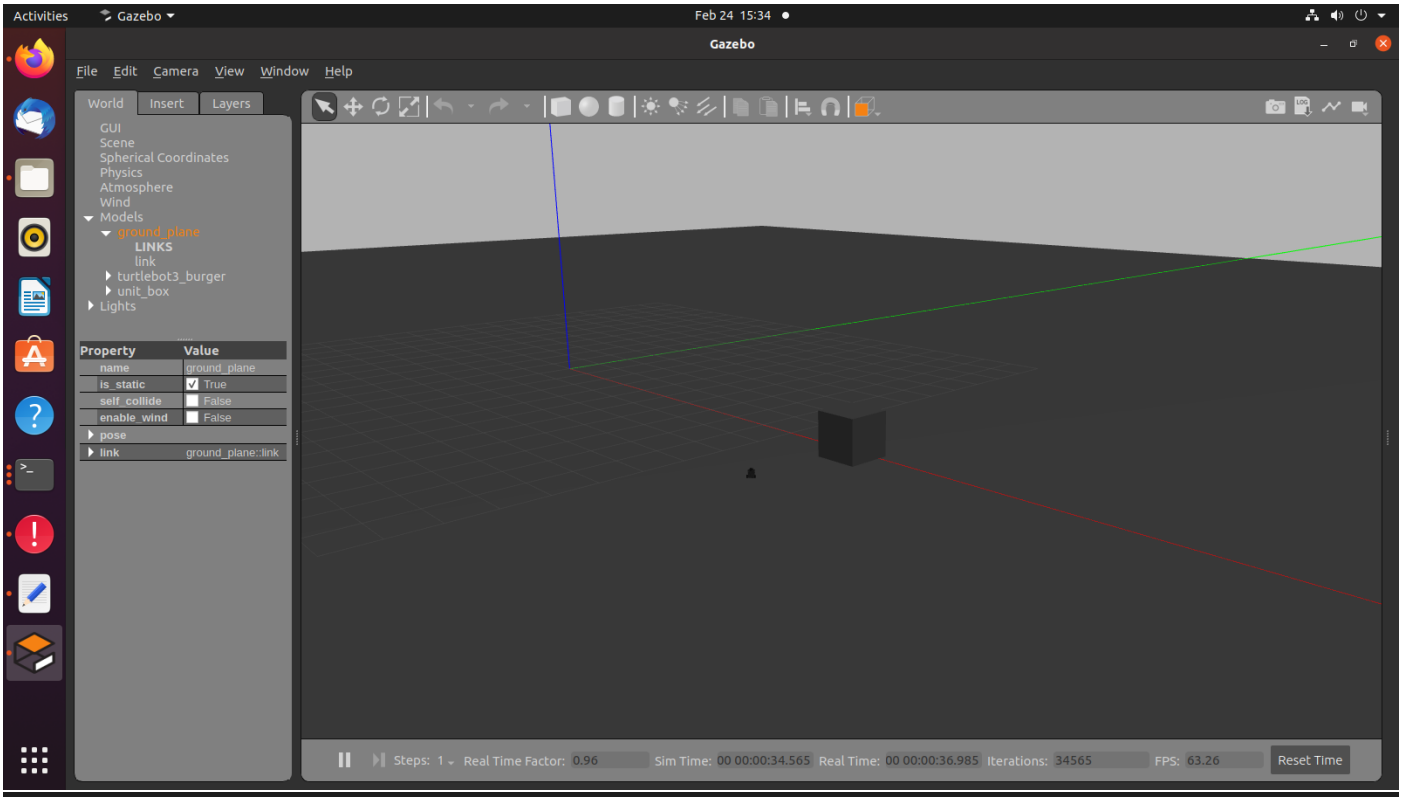

Activities
Gazebo
Feb 24 15:34
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
ground_plane
LINKS
link
turtlebot3_burger
unit_box
Lights
Property Value
name ground_plane
is_static True
self_collide False
enable_wind False
pose
link ground_plane::link
Steps: 1 Real Time Factor: 0.96 Sim Time: 00 00:00:27.424 Real Time: 00 00:00:29.359 Iterations: 27424 FPS: 63.89
Reset Time

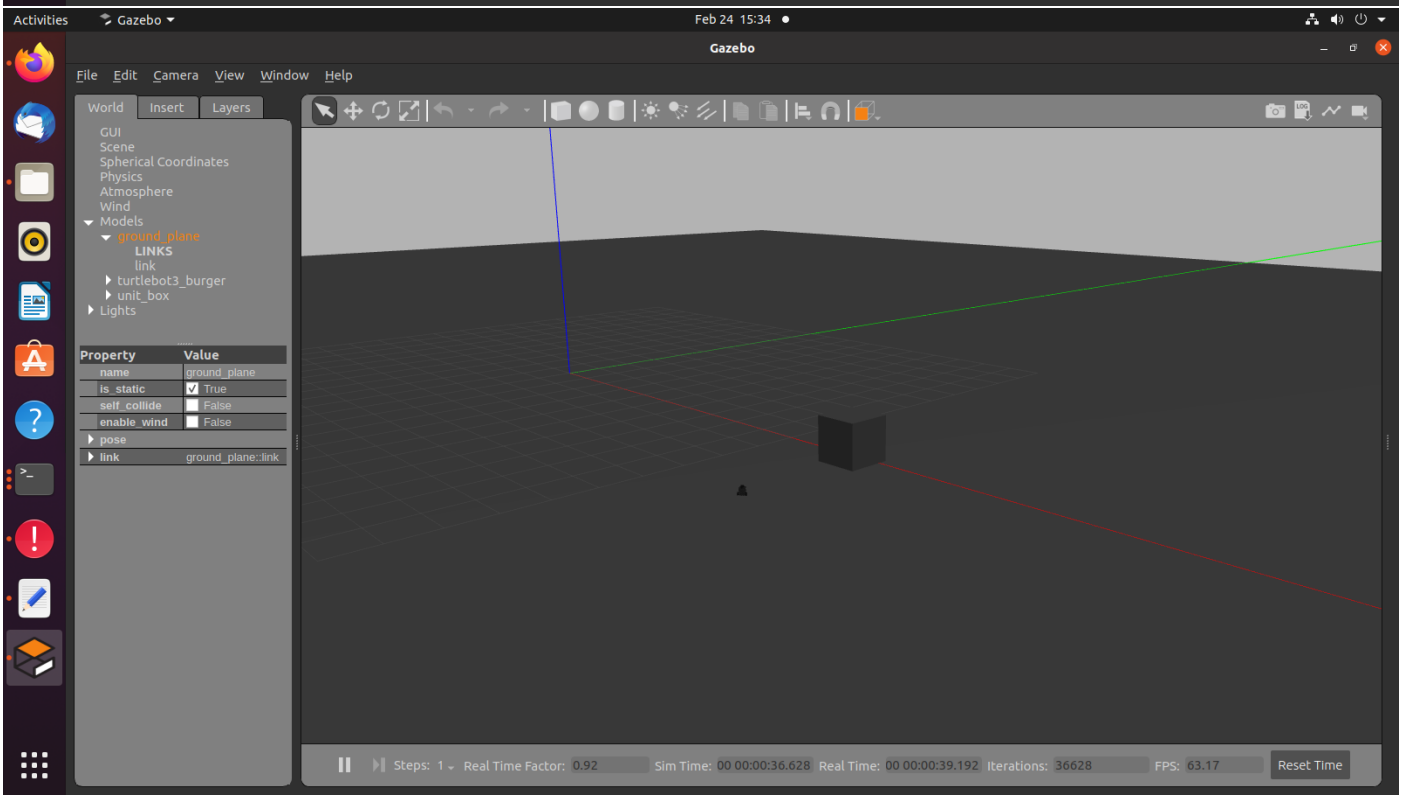

Activities
Gazebo
Feb 24 15:34
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
ground_plane
LINKS
link
turtlebot3_burger
unit_box
Lights
Property Value
name ground_plane
is_static True
self_collide False
enable_wind False
pose
link ground_plane::link
Steps: 1 Real Time Factor: 0.97 Sim Time: 00 00:00:28.744 Real Time: 00 00:00:30.763 Iterations: 28744 FPS: 64.18
Reset Time

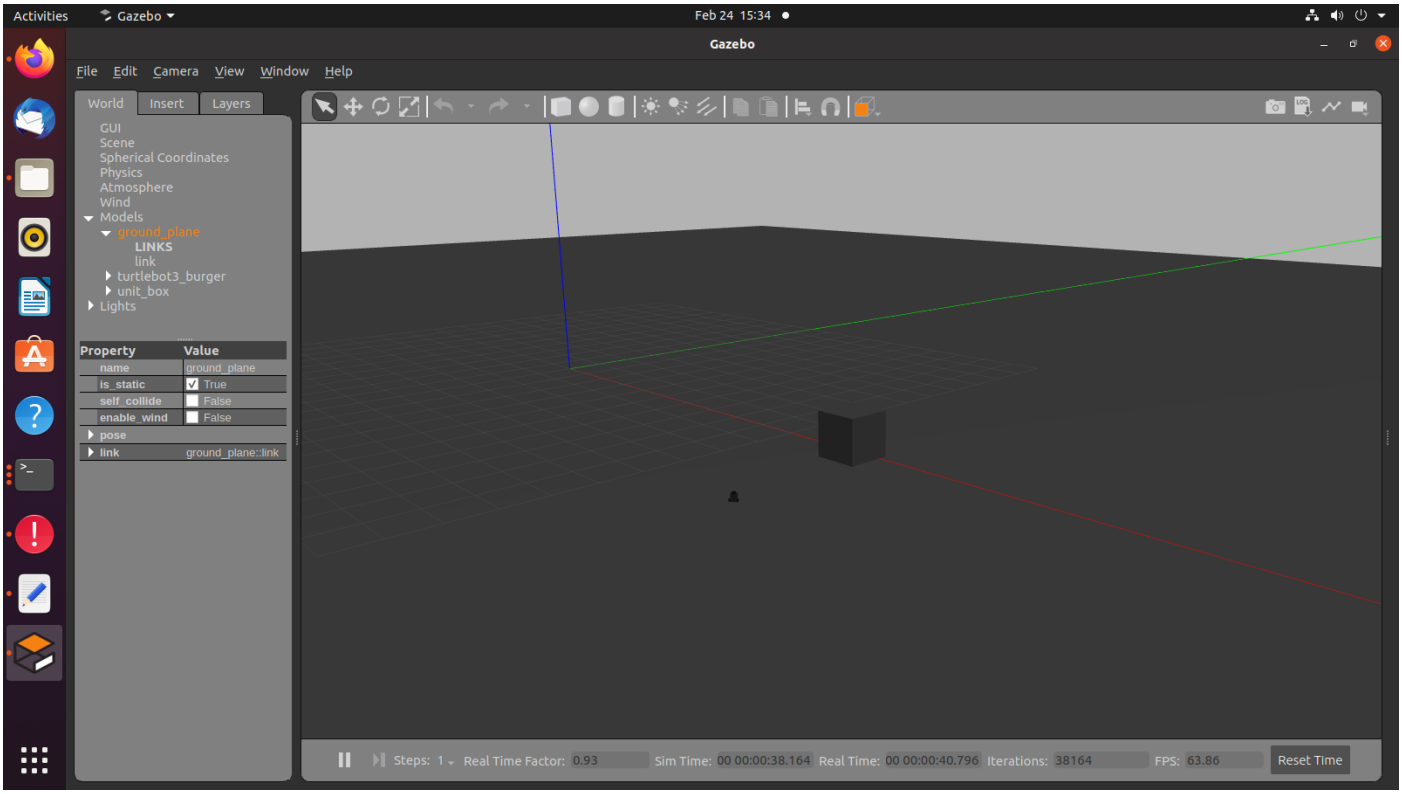

Activities
Gazebo
Feb 24 15:34
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
ground_plane
LINKS
link
turtlebot3_burger
unit_box
Lights
Property Value
name ground_plane
is_static True
self_collide False
enable_wind False
pose
link ground_plane::link
Steps: 1 Real Time Factor: 0.95 Sim Time: 00 00:00:29.994 Real Time: 00 00:00:32.168 Iterations: 29994 FPS: 63.67
Reset Time


Activities
Gazebo
Feb 24 15:34
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
ground_plane
LINKS
link
turtlebot3_burger
unit_box
Lights
Property Value
name ground_plane
is_static True
self_collide False
enable_wind False
pose
link ground_plane::link
Steps: 1 Real Time Factor: 0.92 Sim Time: 00 00:00:31.148 Real Time: 00 00:00:33.371 Iterations: 31148 FPS: 64.07
Reset Time


Activities
Gazebo
Feb 24 15:34
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
ground_plane
LINKS
link
turtlebot3_burger
unit_box
Lights
Property Value
name ground_plane
is_static True
self_collide False
enable_wind False
pose
link ground_plane::link
Steps: 1 Real Time Factor: 0.94 Sim Time: 00 00:00:32.699 Real Time: 00 00:00:34.977 Iterations: 32699 FPS: 63.39
Reset Time


Activities
Gazebo
Feb 24 15:34
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
ground_plane
LINKS
link
turtlebot3_burger
unit_box
Lights
Property Value
name ground_plane
is_static True
self_collide False
enable_wind False
pose
link ground_plane::link
Steps: 1 Real Time Factor: 0.96 Sim Time: 00 00:00:34.565 Real Time: 00 00:00:36.985 Iterations: 34565 FPS: 63.26
Reset Time


Activities
Gazebo
Feb 24 15:34
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
ground_plane
LINKS
link
turtlebot3_burger
unit_box
Lights
Property Value
name ground_plane
is_static True
self_collide False
enable_wind False
pose
link ground_plane::link
Steps: 1 Real Time Factor: 0.92 Sim Time: 00 00:00:36.628 Real Time: 00 00:00:39.192 Iterations: 36628 FPS: 63.17
Reset Time


Activities
Gazebo
Feb 24 15:34
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
ground_plane
LINKS
link
turtlebot3_burger
unit_box
Lights
Property Value
name ground_plane
is_static True
self_collide False
enable_wind False
pose
link ground_plane::link
Steps: 1
Real Time Factor: 0.93
Sim Time: 00 00:00:38.164
Real Time: 00 00:00:40.796
Iterations: 38164
FPS: 63.86
Reset Time