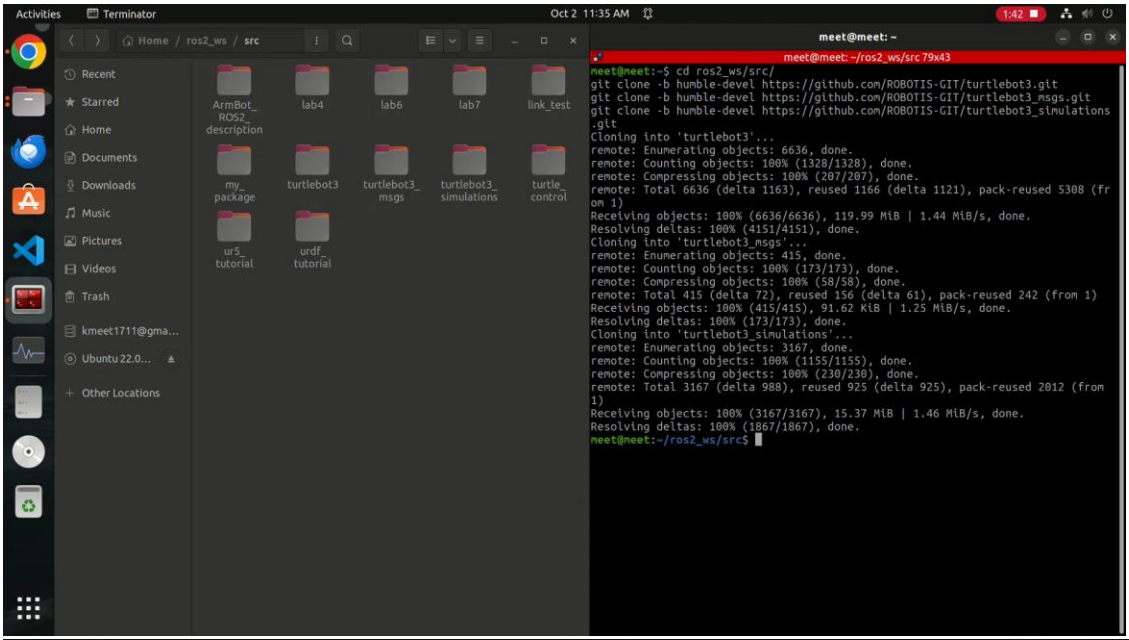
Lab 8 – Exercise

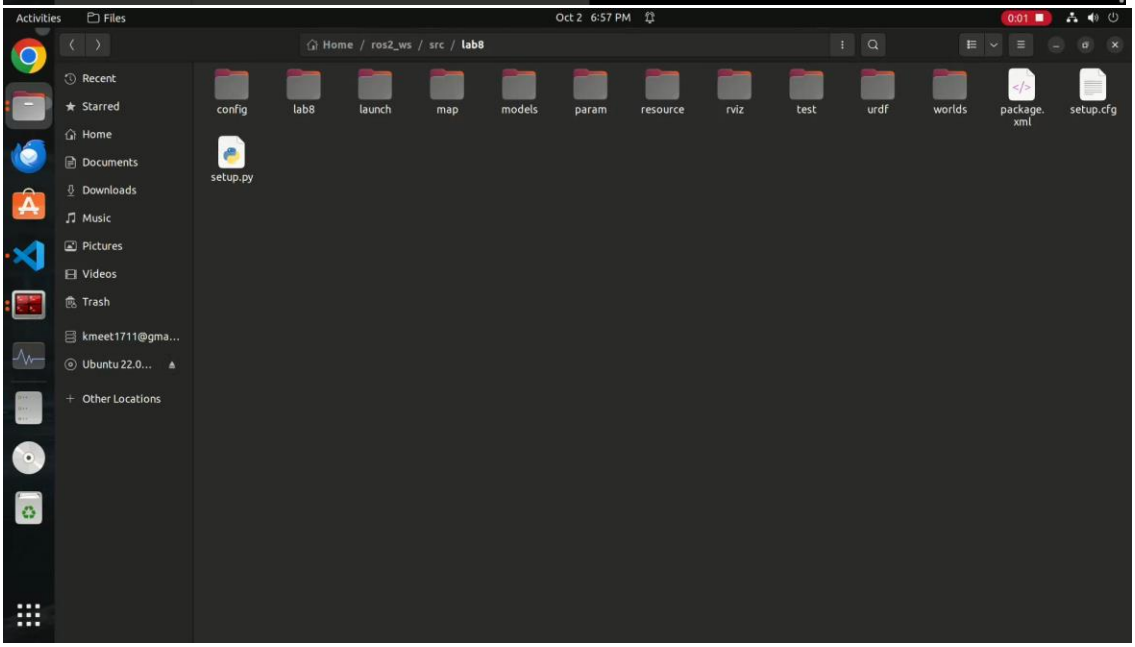
Meet Kansara – 220929270 Roll no. 54

Exercise: To clone the TurtleBot3 repository from GitHub, create a new package with the repository files, generate maps using Cartographer, and navigate TurtleBot3 autonomously on the created maps. Additionally, to create a custom world using Gazebo's building editor, generate a map of the custom world, and navigate TurtleBot3 autonomously in this environment.

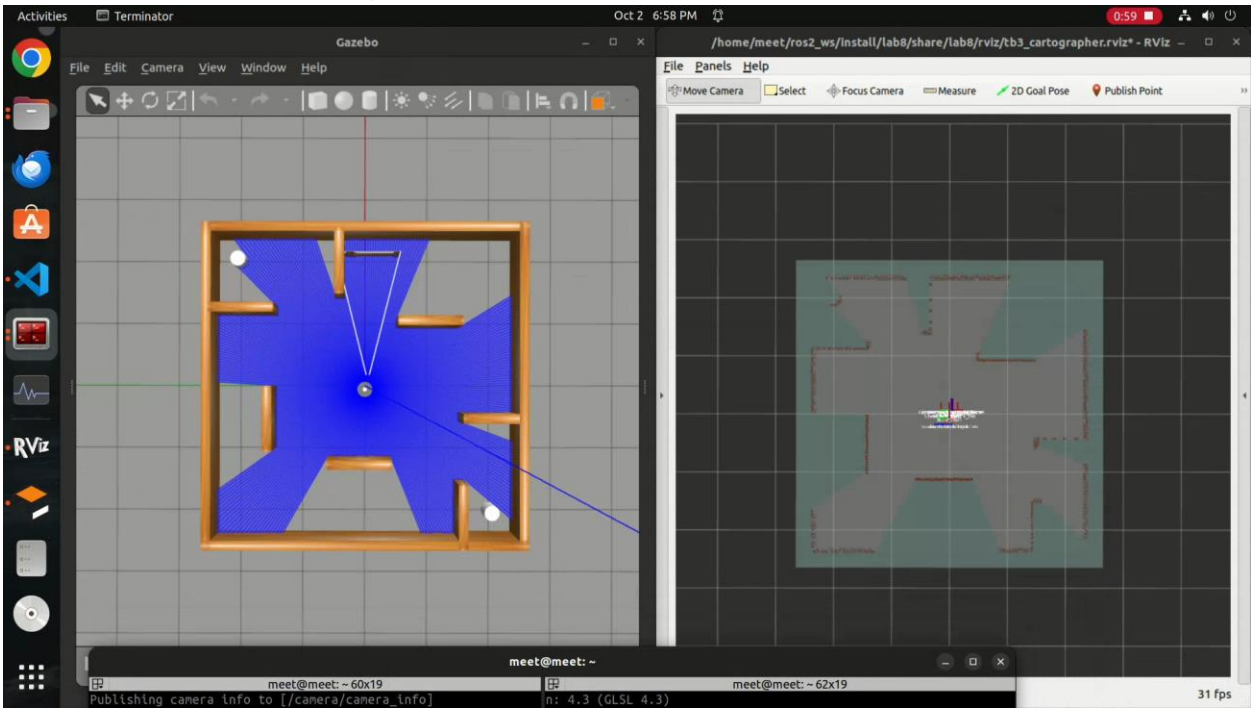
Execution and analysis:

1. Cloning TurtleBot3 repository and creating new package

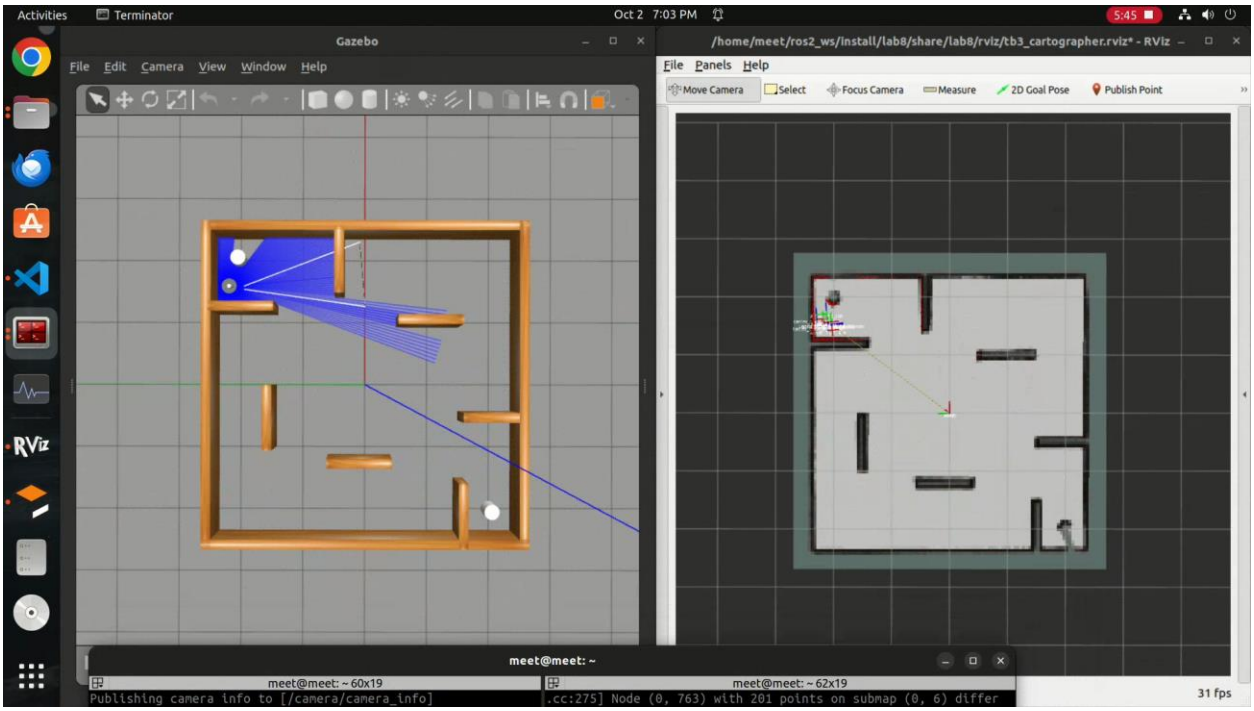




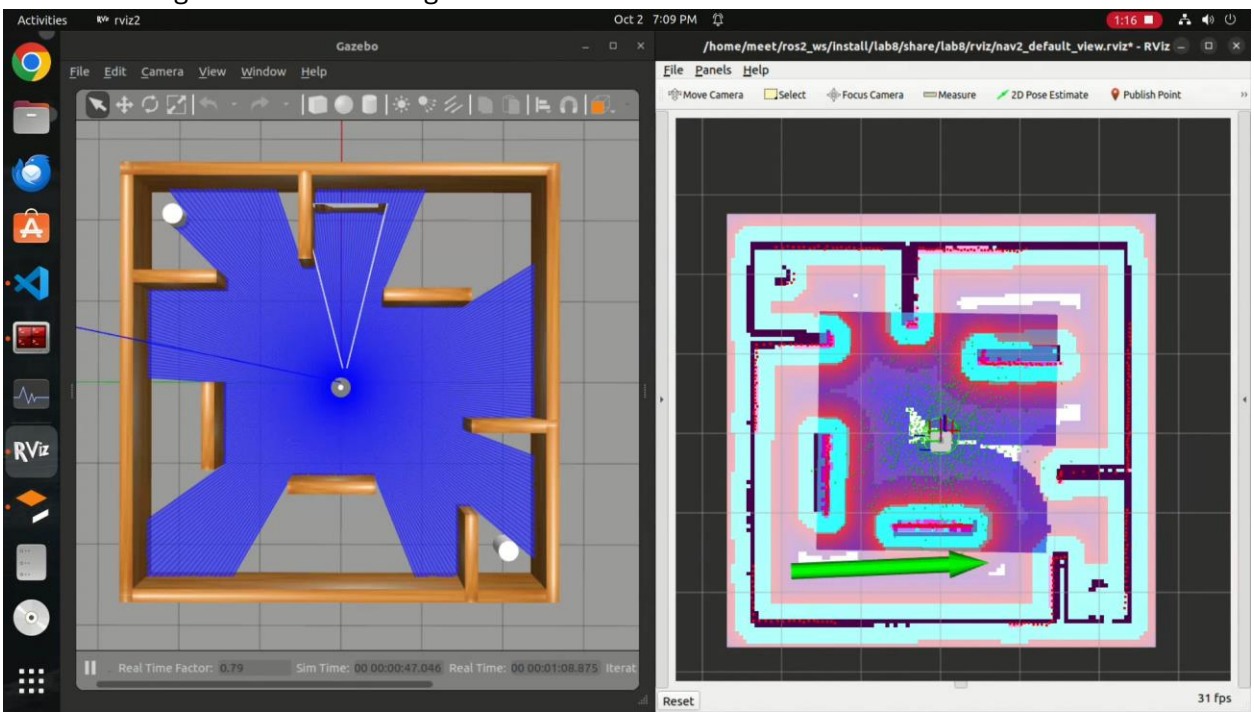
2. Map Creation with Cartographer
    - Initial View Before Map Creation



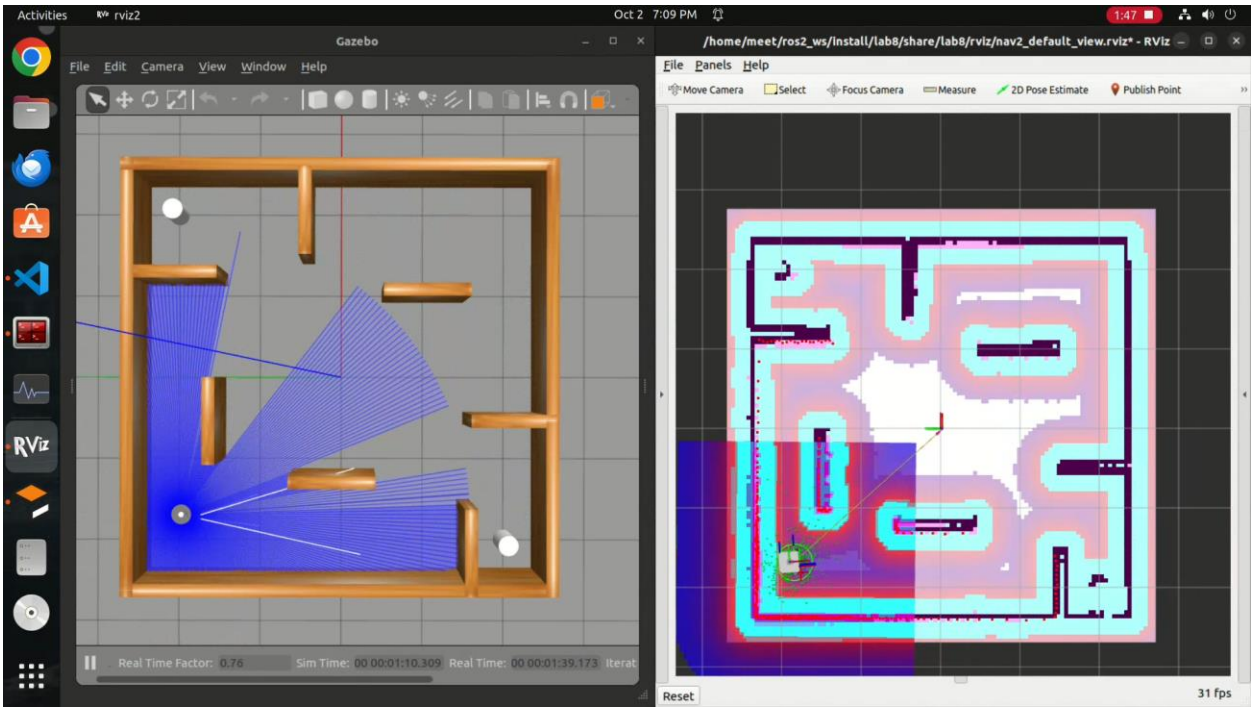
    - Final View After Map Creation



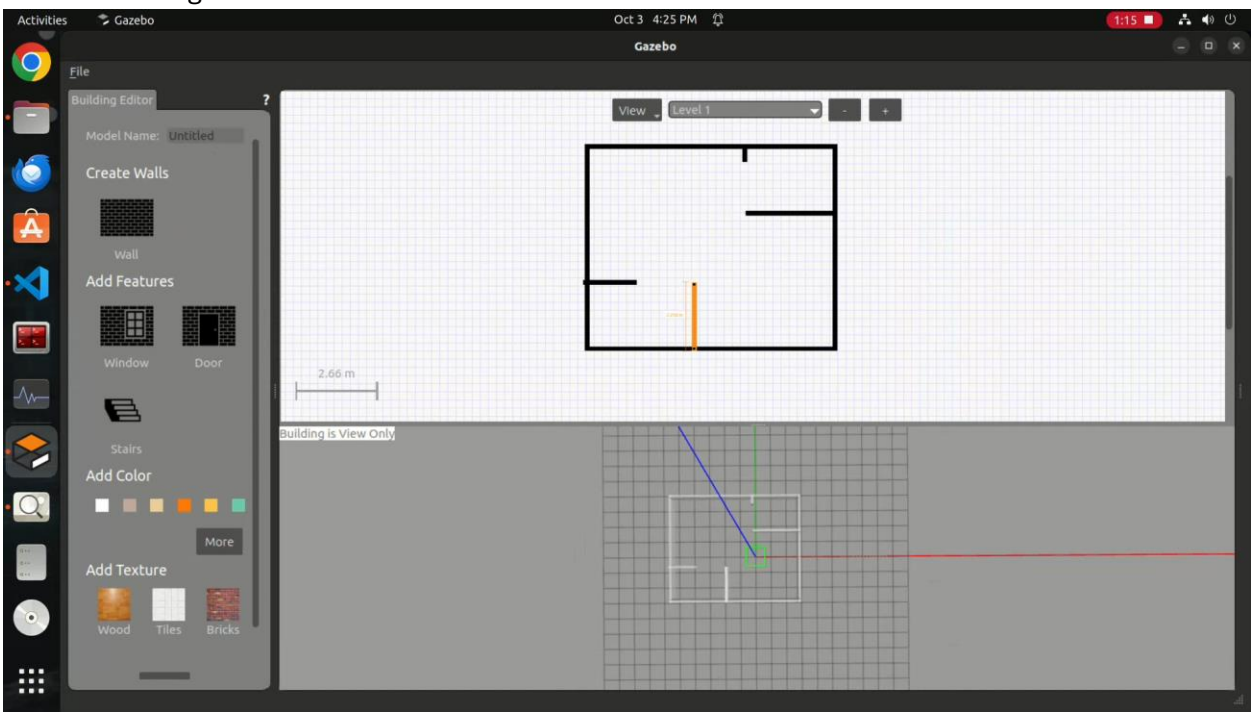
3. Navigation with Cartographer Map
    - Setting Goal Pose for Navigation



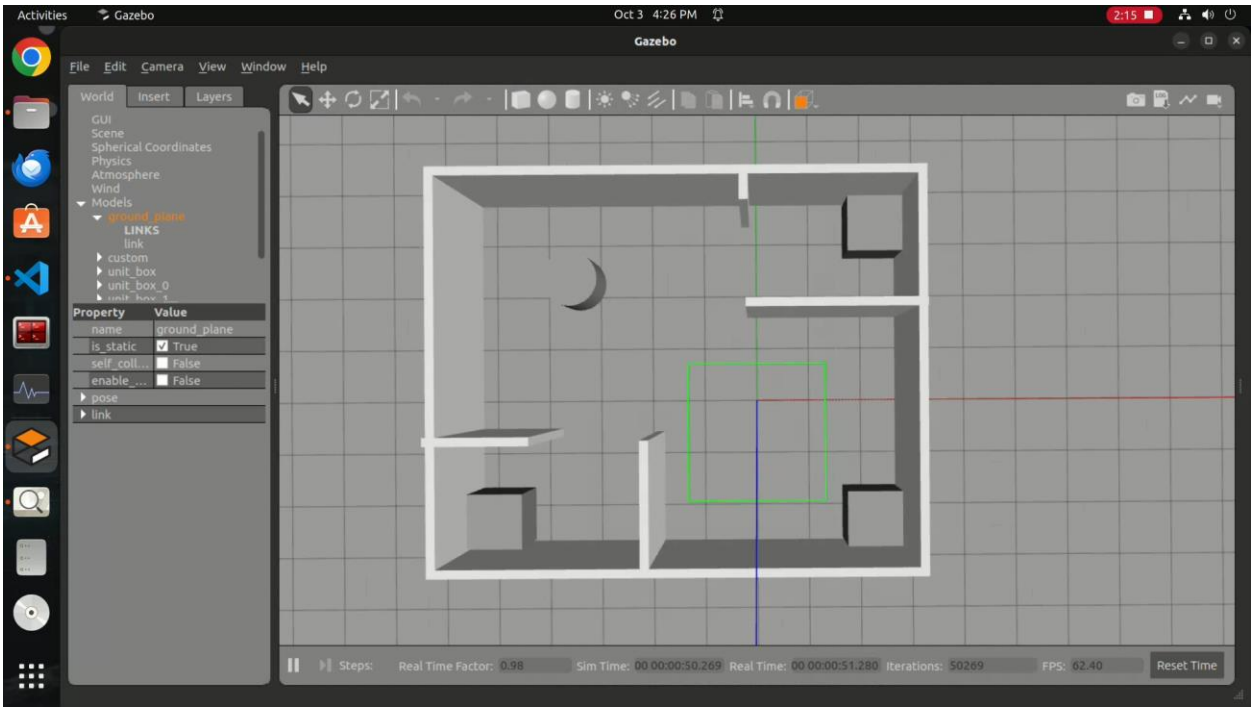
    - Final Position After Navigation



4. Creating Custom World with Building Editor in Gazebo
    - Building editor



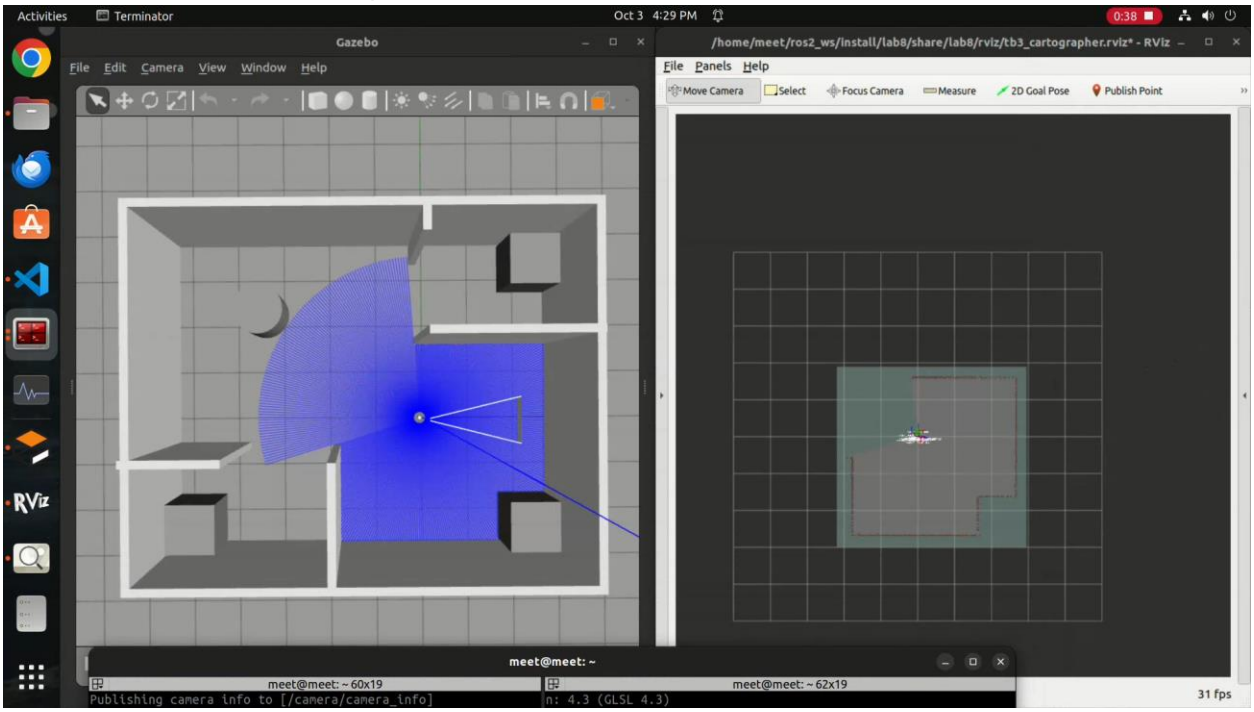
    - Custom world with obstacles



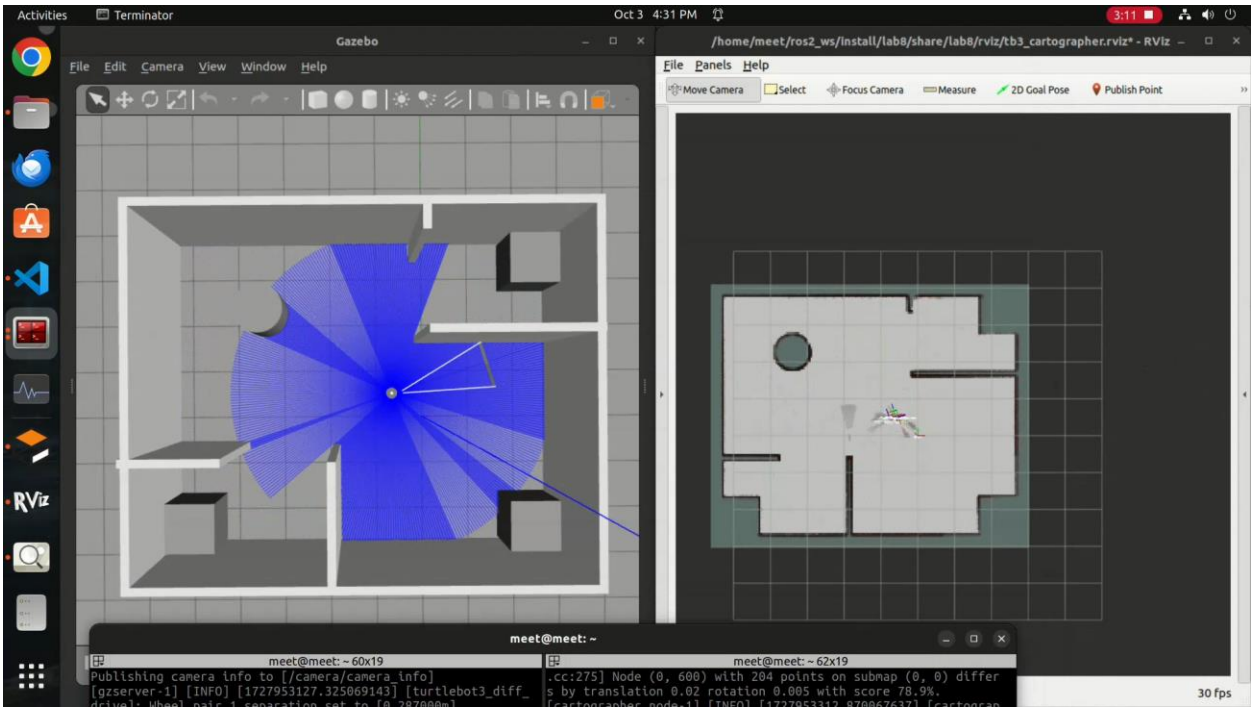
5. Launching Custom World with TurtleBot3

```python
import os

from ament_index_python.packages import get_package_share_directory
from launch import LaunchDescription
from launch.actions import IncludeLaunchDescription
from launch.launch_description_sources import PythonLaunchDescriptionSource
from launch.substitutions import LaunchConfiguration


def generate_launch_description():
    launch_file_dir = os.path.join(get_package_share_directory('lab8'), 'launch')
    pkg_gazebo_ros = get_package_share_directory('gazebo_ros')

    use_sim_time = LaunchConfiguration('use_sim_time', default='true')
    x_pose = LaunchConfiguration('x_pose', default='0.0')
    y_pose = LaunchConfiguration('y_pose', default='0.0')

    world = os.path.join(
        get_package_share_directory('lab8'),
        'worlds',
        'custom.world'
    )

    gzserver_cmd = IncludeLaunchDescription(
        PythonLaunchDescriptionSource(
            os.path.join(pkg_gazebo_ros, 'launch', 'gzserver.launch.py')
        ),
        launch_arguments={'world': world}.items()
    )

    gzclient_cmd = IncludeLaunchDescription(
        PythonLaunchDescriptionSource(
            os.path.join(pkg_gazebo_ros, 'launch', 'gzclient.launch.py')
        )
    )

    robot_state_publisher_cmd = IncludeLaunchDescription(
        PythonLaunchDescriptionSource(
            os.path.join(launch_file_dir, 'robot_state_publisher.launch.py')
        ),
        launch_arguments={'use_sim_time': use_sim_time}.items()
    )

    spawn_turtlebot_cmd = IncludeLaunchDescription(
        PythonLaunchDescriptionSource(
            os.path.join(launch_file_dir, 'spawn_turtlebot3.launch.py')
        ),
        launch_arguments={
            'x_pose': x_pose,
            'y_pose': y_pose
        }.items()
    )

    ld = LaunchDescription()

    # Add the commands to the launch description
    ld.add_action(gzserver_cmd)
    ld.add_action(gzclient_cmd)
    ld.add_action(robot_state_publisher_cmd)
    ld.add_action(spawn_turtlebot_cmd)

    return ld
```

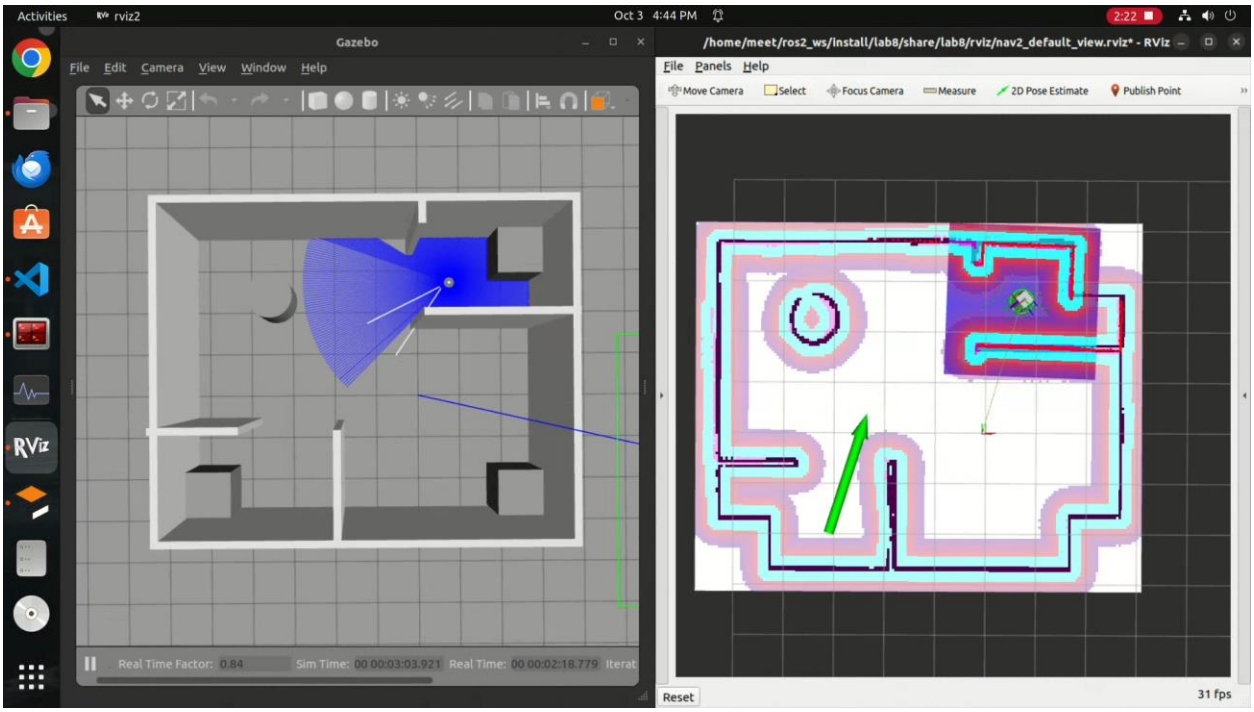
6. Creating Map of Custom World
    - Initial View Before Map Creation



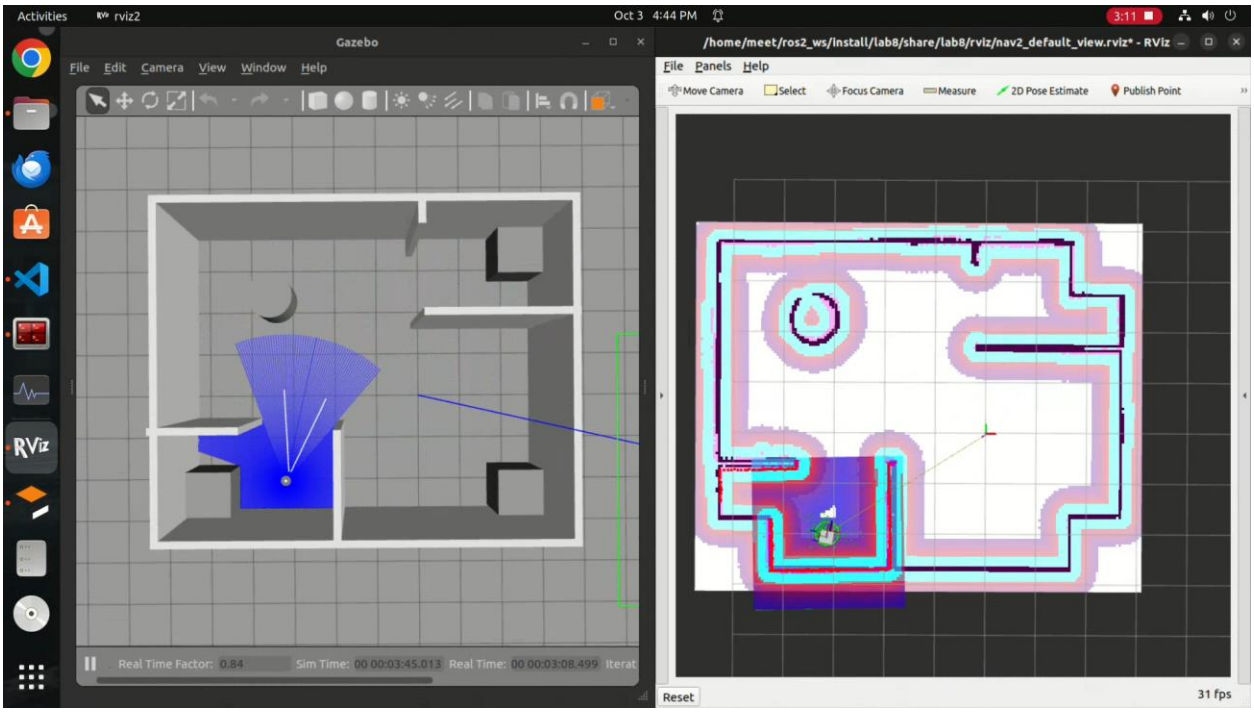
    - Final View After Map Creation



7. Navigation in Custom World
    - Setting Goal Pose for Navigation



    - Final Position After Navigation



Conclusion: The TurtleBot3 repository was successfully cloned, a new package was created, maps were generated using Cartographer, and TurtleBot3 effectively navigated the generated maps. A custom world was also created using Gazebo's building editor, and TurtleBot3 successfully navigated the custom environment.