CTE TechWeekend Fall 2020

Robotics Hackathon

This week-long hackathon will start on **3rd October 2020** (Saturday) **9:00 PM** and will end on **10th October** (Saturday) **9:00 PM**.

This hackathon is aimed to test the best of your technical and logical skills in the field of robotics. To solve the questions, you will require to have a basic understanding of python programming, autonomous navigation, Robotics Operating System (ROS), and perception.

Please go through all the instructions carefully before attempting the questions. All the materials required for the hackathon are available in this Github repository.

Please keep checking the repository throughout the week for any kinds of updates or corrections.

Note: We understand the complexity of the questions provided for the hackathon and thus encourage any kind of partial solution that you can come up with. Winners will be decided on the basis of closeness to complete the task and the efficiency of the solution.

Contact:

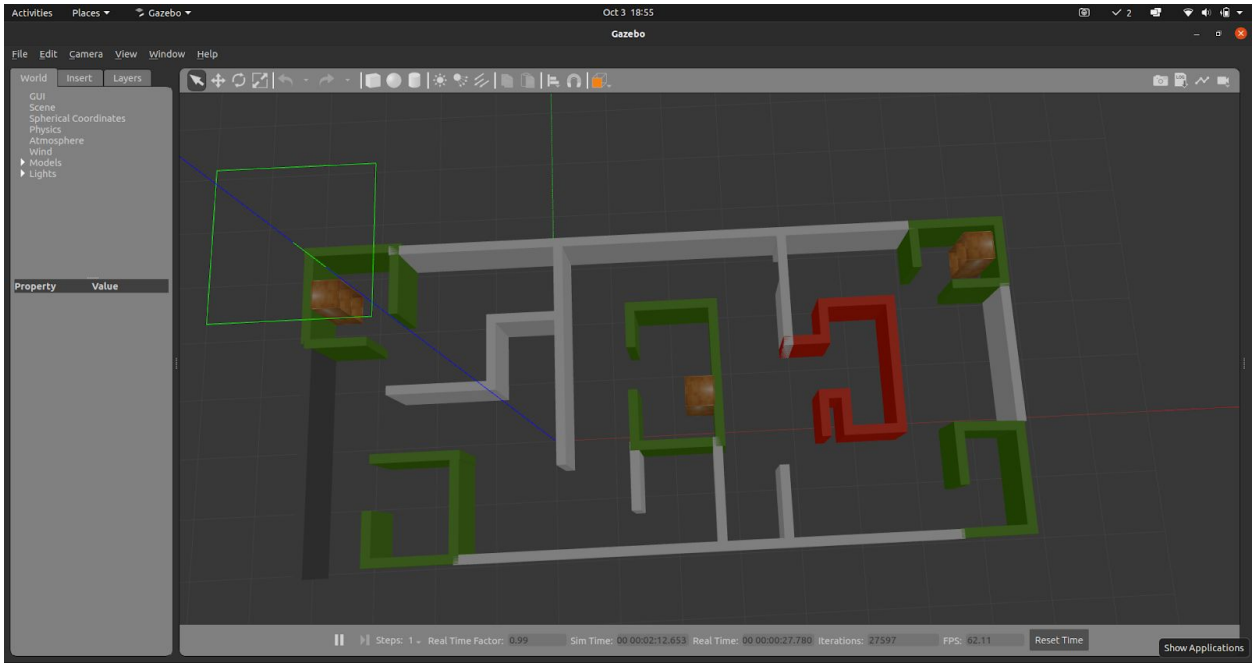
- Please join the slack channel with this LINK. It will be a place for announcement and doubt clarification,
- You can also contact:
    - Suhrudh : +91 93902 62933,
    - Pranav : +91 9307894704,
    - Vedant : +91 7359313678

Problem Statement

The problem statement of the hackathon is divided into two parts, both the parts are independent of each other and need to be separately solved. The world file required to solve both parts is the same and is included in the Github repository.

- Part A

The maze :



In the world file present in the Github repository, you have been provided with a maze. You need to create a ROS Package to **autonomously** navigate a robot from the entrance of the maze(refer Fig 2) to the region marked with red walls inside the maze(to Target Area (refer to Fig 1)). You are allowed to use any kind of path planning and navigation algorithm you

feel appropriate. The robot used in the task should strictly be a ground terrain robot. You can use existing turtlebots like Waffle or Waffle Pi or make your own customized one. You are allowed to use any kind of sensors or equipment on your robot.

Fig 1

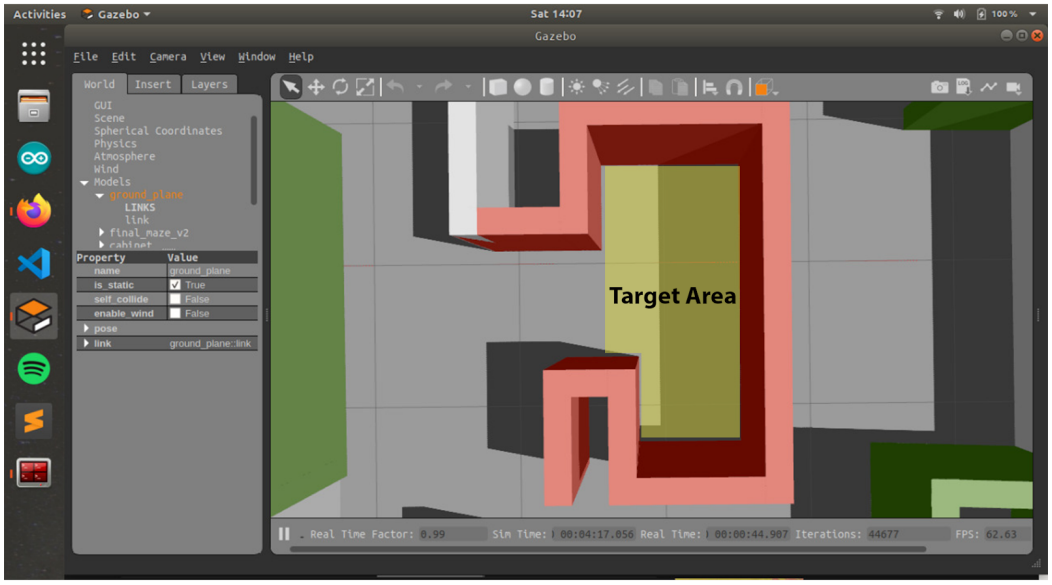
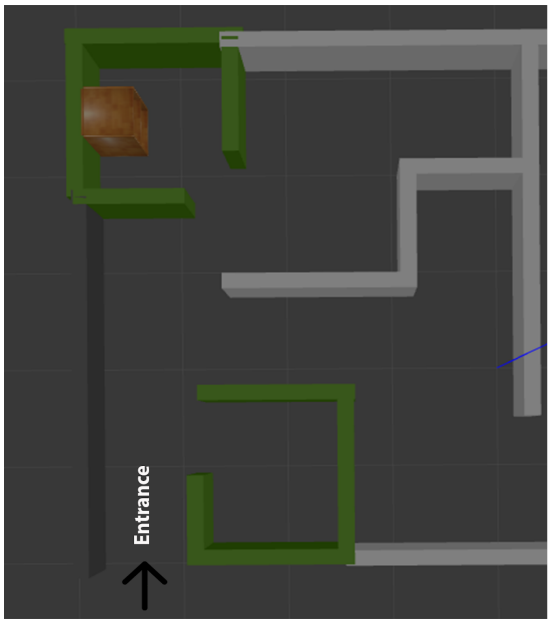


Fig 2



Rough odomentry value for starting point:

position:

x: -2.48769792484e-05
y: 3.30728333069e-05
z: -0.00100139297939

orientation:

x: -8.04851441217e-06
y: 0.00385309979396
z: 0.000256124010198
w: 0.999992543951

Rough odometry values for the target region:

position (anything inside 0.5m radius of the given value):

x: 8.45798701012
y: 2.90261823569
z: -0.00100145045392

orientation :

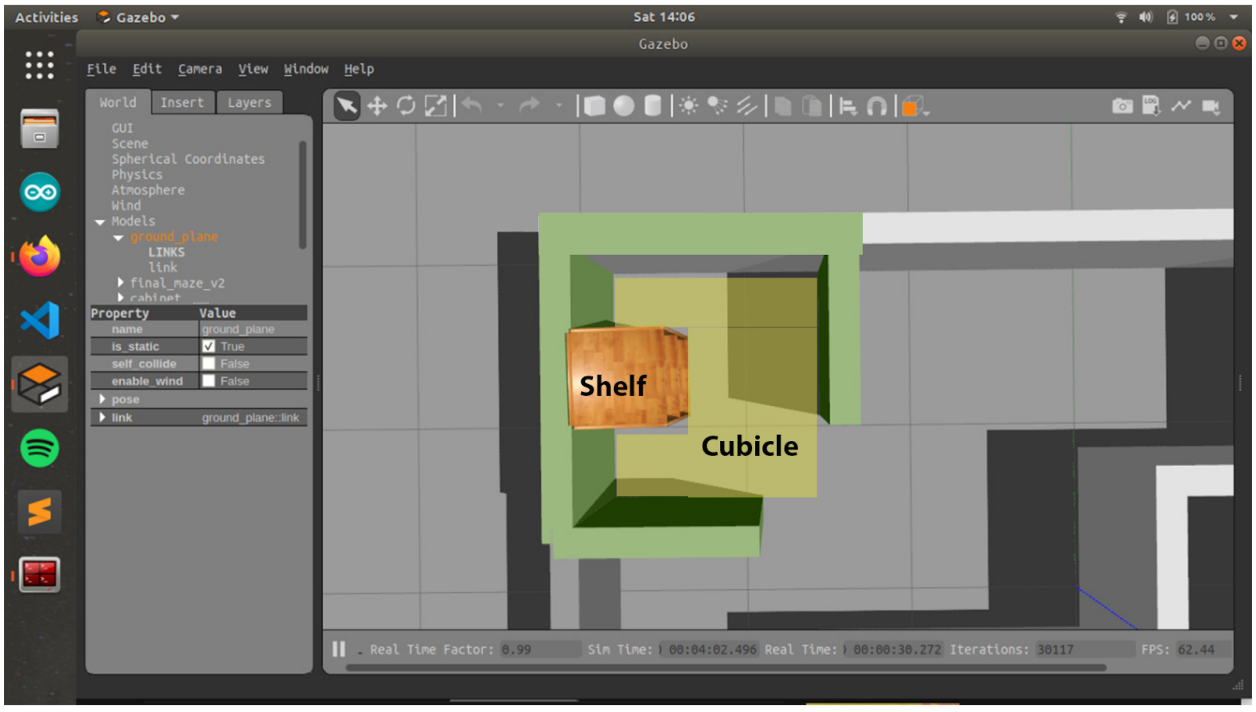
x: -4.16323870161e-05
y: 0.00385339712874
z: 0.00907028220335
w: 0.99995143861

Note: Both the odometry information are derived from the default location of the maze inside the world file, you are allowed to shift the maze wherever you feel convenient but remember to transform the odometry corresponding to that location of the maze

- Part B

In the same maze as given for PART A, there are some cubicles(refer Fig 3) present throughout the maze. These cubicles are marked with green walls on all sides. They also contain an object(cabinet).
Your task is to find the number of such cubicles by **autonomously** exploring the maze through your robot and publish the total number of such cubicles explored as a ROS message. You will be required to use color detection to give identification of the cubicle to your robot. Having a cabinet is also a unique feature of these cubicles that you will need to use to explore. Any kind of ground robot with an onboard camera is allowed for this task (Waffle Pi, Waffle, etc). You are allowed to use any kind of sensors and equipment on the robot.

Fig 3



Submission Guidelines

1. You are required to submit:
    - A Github repository containing your package for the hackathon
    - A short one-page summary describing your methodology
    - Set of instructions to run and reproduce the behavior of the bot (This can include which commands to run, which packages to install, launch files, etc).

2. Ensure that your repository is kept private until the end of the contest
3. **Note**: Conditions for Disqualification.
    a. If your repository is not kept private throughout the hackathon.
    b. If you are found copying from other competitors.
    c. If it is found that your robot is found to be operated using teleoperation.
    d. Or any other grounds that may be deemed as plagiarism.

If you have any doubts, please feel free to raise them in the slack channel.

We hope you have an amazing experience and get to learn lots of new things while working on this Hackathon.
All the Best!!