PROJECT 3A: PURSUIT EVASION GAME (PART-1)

Tanmay Khandait
Student ID : 1219385830
CIDSE, Arizona State University
tkhandai@asu.edu

09 April, 2021

ABSTRACT

This report being submitted for Project-3a along with the necessary resources. This project is an introduction on how to control robots in Gazebo and Rviz in order to build our own robot models as well as environments to perform some experiments involving navigation, localization etc.

1 Build a scene using Gazebo

The idea here is to construct an scene/environment in Gazebo whihc can be used for the purpose of simulation. For this project, we were asked to create an environment of a house.

For my environment, I chose to model the dorm flat that I used to live in during my undergraduate program[1]. I started by building the skeleton of the world (walls, doors and windows) using the model editor. Once this environment was ready, it was time to add some objects to the house. We have very limited options (i.e, a cube, sphere and cylinder) if we use the deafult Gazebo objects. In order to add some real-life feel to it, I added objects like bed, sofa, table and chairs from the AWS-Robomaker github repository[2]. The various stages of scene building and the final models are shown in figure 1.

The codes for this have been adapted from [3]. The adaptation which was done was for launch files and to understand how to launch the environment in gazebo.

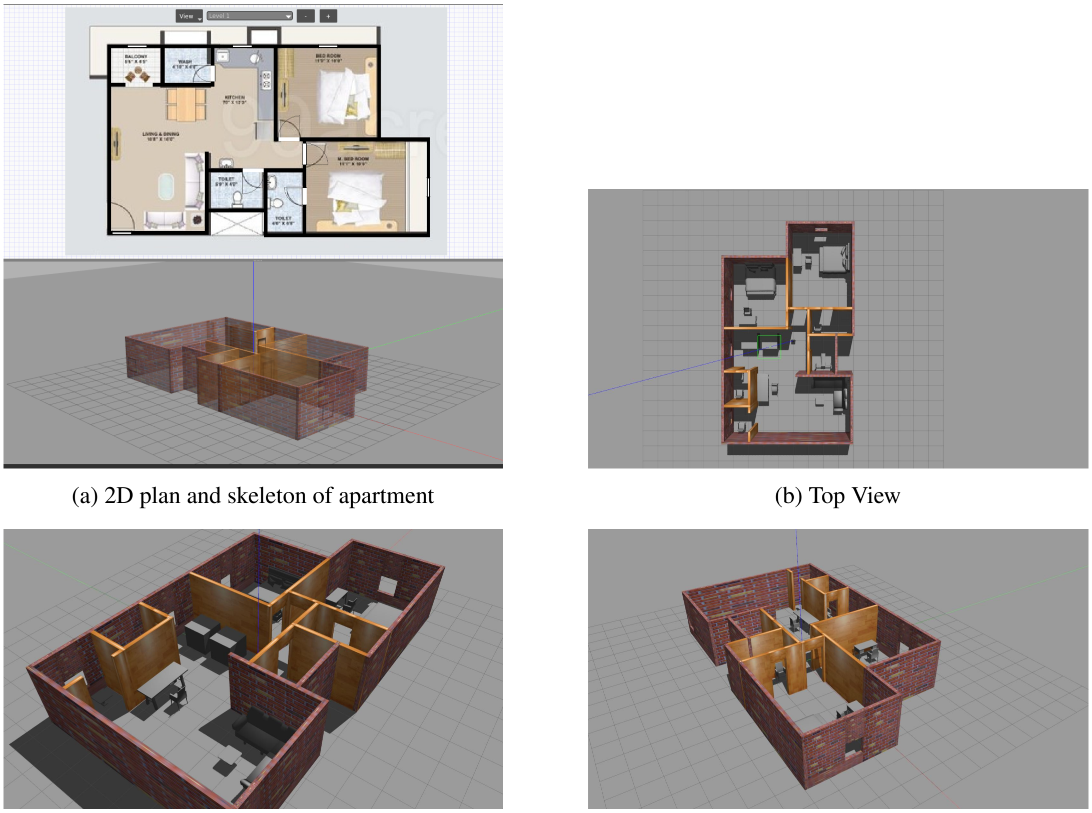
2 Operate a Turtlebot3 robot in the scene

In this task, the aim is to spawn the robot in every room and ensures that robot visits every room and draws the first inital of last name (in my case, it will be the letter K). I approached the problem in the by first investigating and playing around with robot. I wanted to understand how the robot works, how the navigation and works in order to work on the task.

There were two approaches possible. The first one was to manually let the robot go around in every room and then draw the initial. The second approach was to use some sort of automation to move the robot to every room and then start a script that can draw the initial. The first approach seemed like a huge task. I went with the second approach.

For this approach, I used the SLAM tutorial from the robotis documentation[4]. I launched the SLAM node and started the teleoperation in order to map the environemt of the house. The map that was obtained is show in figure 2. Using this environemnt, I made a python script that uses move_base. I launched the client and then used the MoveBaseGoal in order to send the goal to the robot. This ensured that the robot went to certain coordinates. In order to find the coordinates, I used the odometry message and moved the robot around to every room. This allowed me to find the coordinates of every room. Once this was done, I took the script from Project_1a to draw the initial of my letter. This script basically used the cmd_vel to move the robot in a certain direction and turn it at a certain angle.

So the overall idea is to move the robot to certain coordinates and launch a script that draws the initial.



(a) 2D plan and skeleton of apartment

(b) Top View

(c) Isometric View 1

(d) Isometric View 2

Figure 1: Different views of the environment that was created.

3 How to run the code?

This code was run Ubuntu 18.04 and Ros-Melodic. The packages are needed are standard in the sense that we need the ROS packages and packages for turtlebot3 installed alraedy. This should satisfy the requirements.

- Open the terminal window and paste the following

```
roscore
```

- Open a new terminal window and paste the following

```
source ~/proj3a_ws/devel/setup.bash
export TURTLEBOT3_MODEL=waffle_pi
roslaunch world_simulation my_world.launch
```

  This will launch gazebo with the environment and the robot spawned in.

- Open a new terminal window and paste the following

```
source ~/proj3a_ws/devel/setup.bash
export TURTLEBOT3_MODEL=waffle_pi
roslaunch tb_navigation turtlebot3_navigation.launch
    map_file:=~/proj3a_ws/src/tb_navigation/maps/map.yaml
```

  This will launch the rviz window with the map obtained from SLAM and the robot spawned in.

- Open a new terminal window and paste the following

```
source ~/proj3a_ws/devel/setup.bash
chmod +x ~/proj3a_ws/src/world_simulation/scripts/moveBot.py
rosrun world_simulation moveBot.py
```

  This will start the python script that moves the robot to every room and then draws the initial K.

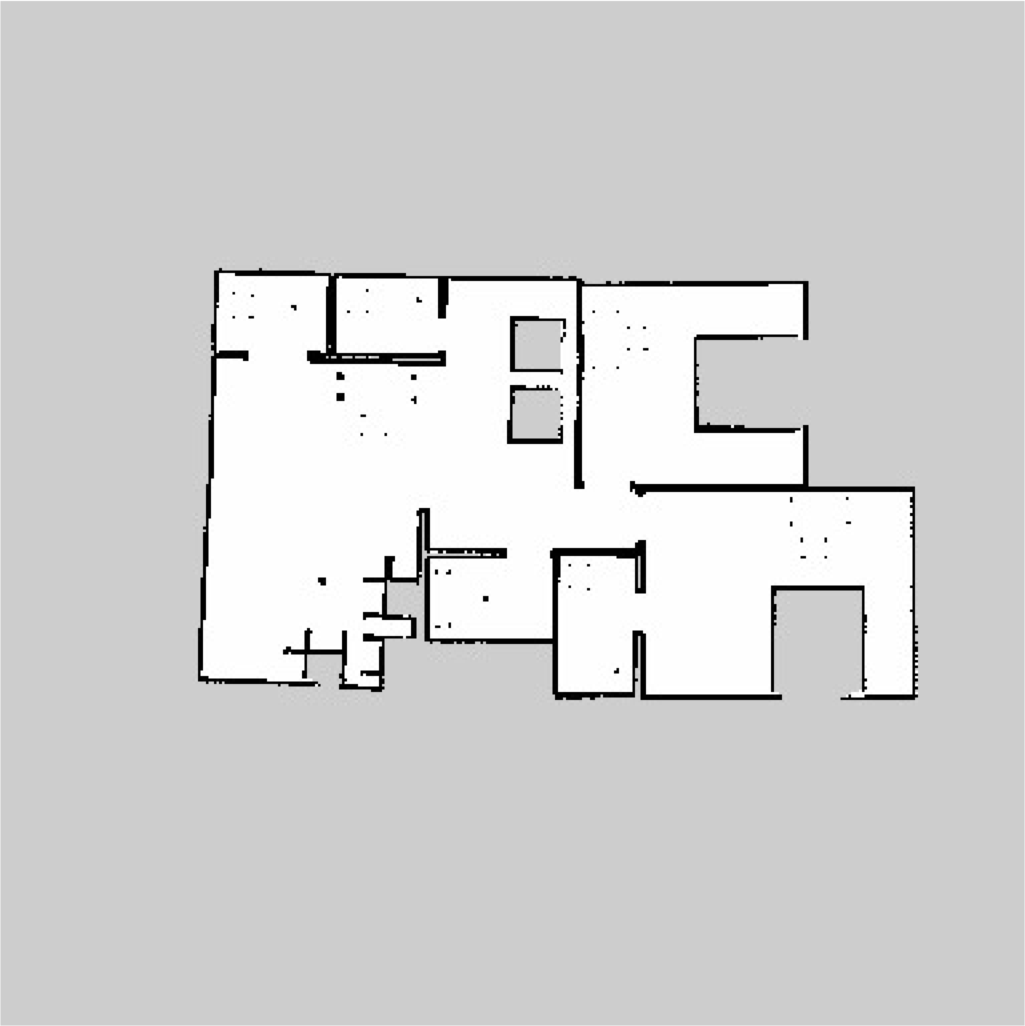


Figure 2: Map generated using SLAM from [4]

4 Acknowledgement

Most of the code was inspired or adapted from [3, 4]. The recording was done using zoom screen recording functionality. The original video was 12.55 mins long. The videos was sped up by 7 times in order to meet the submission criteria (now 1.51 mins long).

Here is the link to small, sped up video: `https://drive.google.com/file/d/1jp5pU_1uTGa_T44WDOiYjThWeCXKkXwf/view?usp=sharing`.

Here is the link to the video folder containing both original and edited videos: `https://drive.google.com/drive/folders/1XwOOD45G2bQIOsKdro4JDh9oGIP4uoM3?usp=sharing`.

References

[1] Shreerang apts 2d plan. `https://newprojects.99acres.com/projects/shree_rang_group/shree_rang_icon/maps/2bhk_121_sq_yd.jpg`. Accessed: April 09, 2021.

[2] Aws robomaker small house world ros package. `https://github.com/aws-robotics/aws-robomaker-small-house-world`. Accessed: April 09, 2021.

[3] Gazebo ros demos. `https://github.com/ros-simulation/gazebo_ros_demos`. Accessed: April 09, 2021.

[4] Robotis turtlebot3. https://github.com/ROBOTIS-GIT/turtlebot3_simulations. Accessed: April 09, 2021.