Map Generation Tutorial

1. Introduction

In this tutorial, we shall present a methodology for the generation of new maps. This enables you to employ these newly created maps during the process of collecting expert demonstrations. The integration of demonstrations originating from a variety of maps will contribute to the enhancement of the robustness and effectiveness of your trained model.

2. Method

Before starting, download the latest version of the source code from our project repository(https://github.com/rllab-snu/Intelligent-Systems-2023).

git clone git@github.com:rllab-snu/Intelligent-Systems-2023.git

(Latest update: 2023/11/08)

2.0. Make a new virtual environment

To generate new maps, you need a new virtual environment that contains Python3.

```
conda create -n py36 python=3.6
```

After creating a new environment, activate it

```
conda activate py36
```

Move to the folder where the source codes exist.

```
cd catkin_ws/src/Intelligent-systems-2023/sim2real/scripts/map_generation
```

and install required packages with pip.

```
pip install -r  requirements.txt
```

2.1. Drawing the walls manually

Currently, there are 3 maps in your environment. You can check in

catkin_ws/src/Intelligent-systems-2023/sim2real/world_map/

Navigate to catkin_ws/src/Intelligent-systems-2023/sim2real/scripts/map_generation folder and check where you are.

```
pwd
```

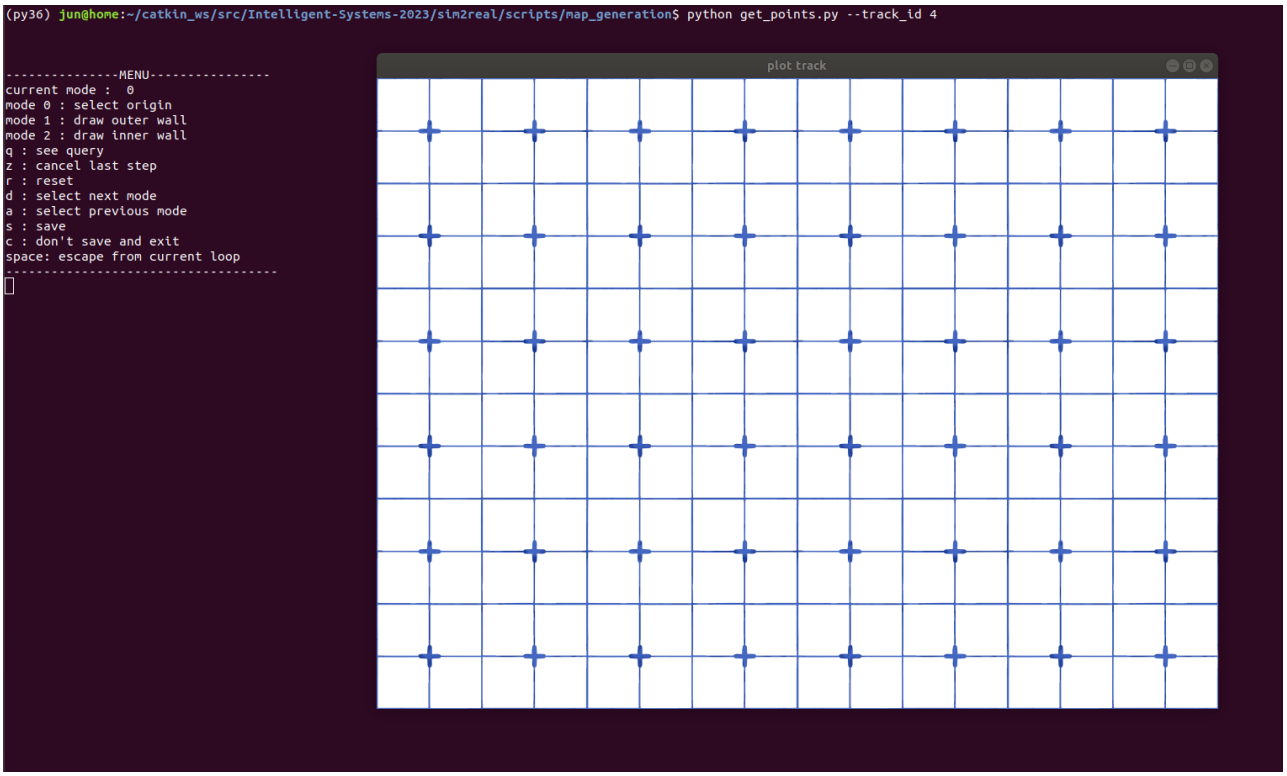
The output should be
/home/{USERNAME}/catkin_ws/src/Intelligent-Systems-2023/sim2real/scripts/map_generation

Choose the id that you want to give to a new track and run the script with command

```
python get _points.py --track_id {ID}

ex) python get_points.py --track_id 4
```

Then you can see the message in the terminal and the grid image .

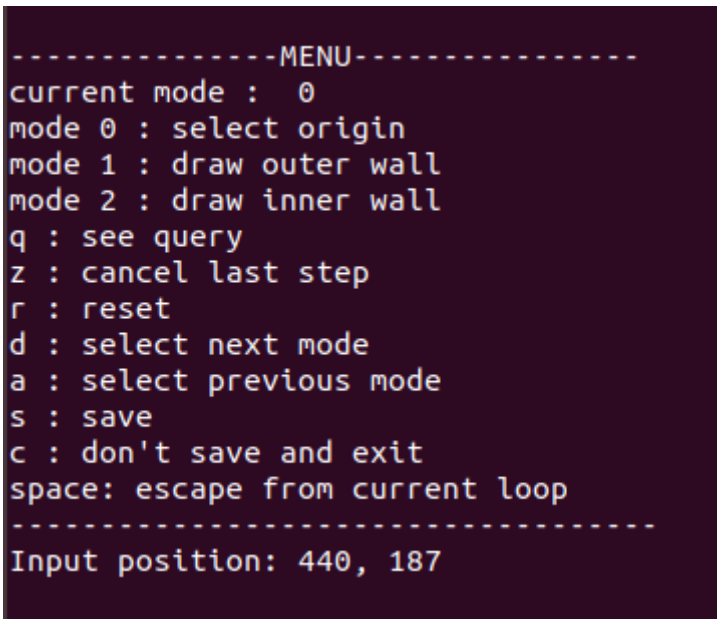


If the picture is too large or too small, you can adjust it with additional arguments, width_ratio and height_ratio. The default values of those arguments are both 0.2

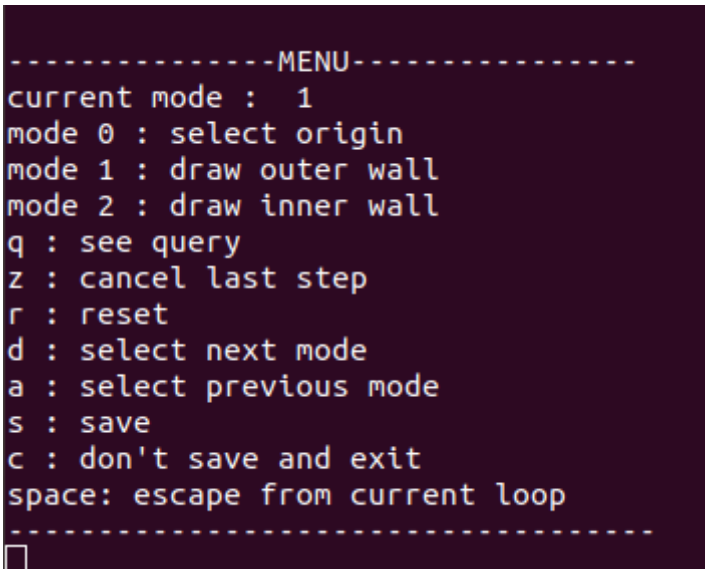
```
ex) python get_points.py --track_id 4 --width_ratio 0.3 --height_ratio 0.3
```

There are 3 modes, select origin, draw outer wall, draw inner wall and these matches with mode 0, 1 and 2, respectively.

By clicking the point you want in the picture, you can select a starting point of a track where the car starts. You can see the message like “Input position: {x,y}”.

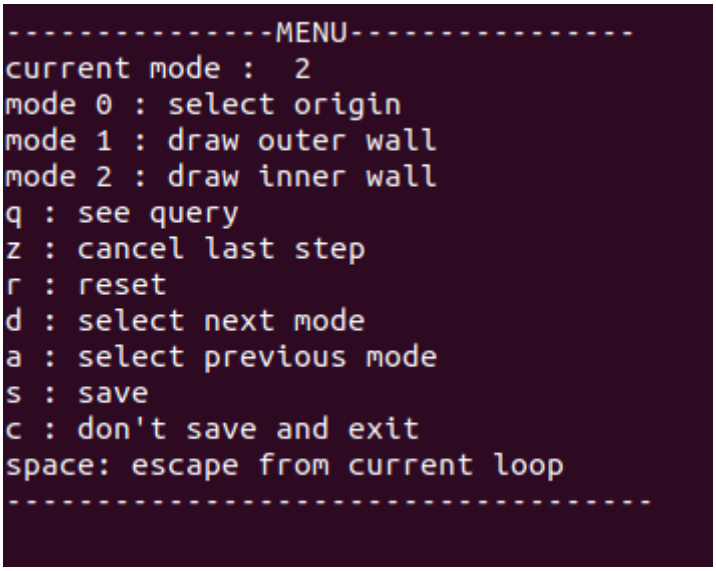


After clicking the position, press ‘d’ to draw the outer wall. Then you can see ‘current mode’ changes to 1 from 0.



Here the loop for connecting the points that you clicked is running, so keep clicking the points to construct the outer wall in the shape you want in the picture. The points are not rendered in the picture, so believe in yourself.

After clicking all the points you want, again press ‘d’. Then you can see the mode is changed to 2, which means now you can draw the inner wall.



Repeat the clicking procedure for the inner wall as for the outer wall. After clicking all the points you want, press ‘s’ to save.

And finally, press ‘c’ to escape.

You can find your result in catkin_ws/src/Intelligent-systems-2023/sim2real/world_map/.

There will be a file named ‘track_{ID}.txt’ where the ID is the argument you chose before.

2.2. Build the world map

Now deactivate the virtual environment with python3, and return to the environment with python2, where the ROS has been installed.

```
conda deactivate

conda activate py27
```

Run the below command to make essential files to construct a full map.

```
python map_generator.py --track_id {ID}
```

You can adjust the scale of the map with an additional argument, ‘scale.’ The default value of scale is 1.0.

```
ex) python map_generator.py --track_id 4 --scale 2.0
```

Then you can check the files with the names ‘track_4.world’ and ‘track_4_scaled.txt’ are generated in …/sim2real/worlds/worlds folder and …/sim2real/worlds/world_map_scaled, respectively.

2.3. Making waypoints

Now visualize your new world that you drawed before with gazebo.

Navigate to …/sim2real/worlds folder. Check where you are. The output should be

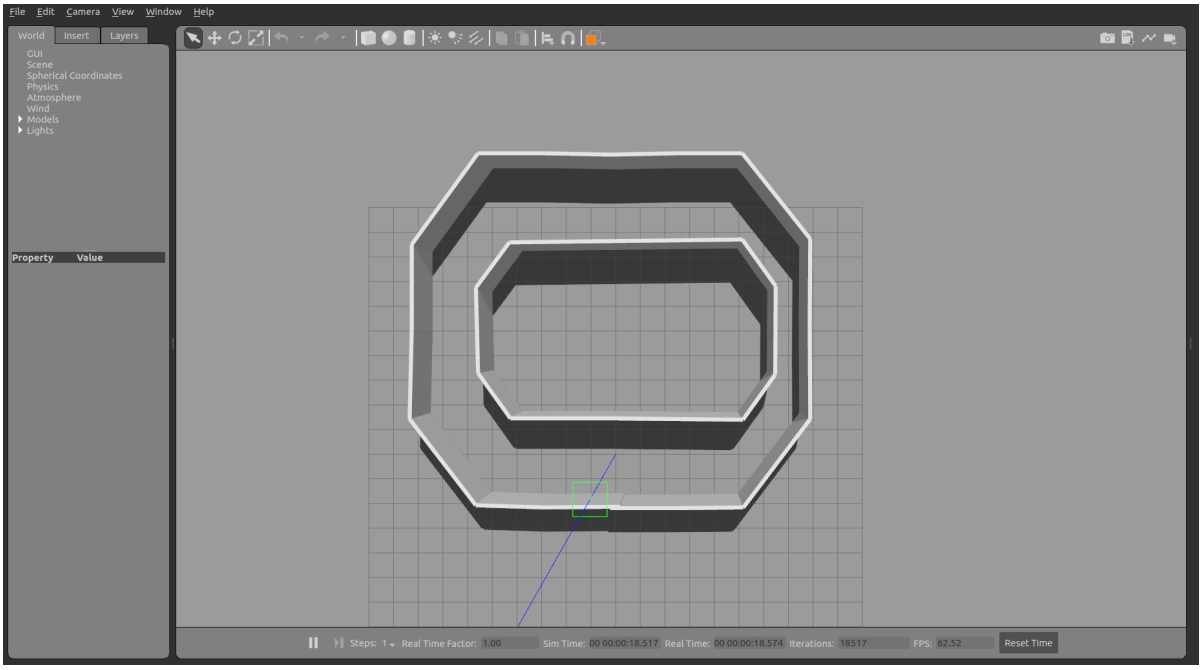
/home/{USERNAME}/catkin_ws/src/Intelligent-Systems-2023/sim2real/worlds

```
pwd
```

Run the following command.

```
gazebo worlds/track_{ID}.world

ex) gazebo worlds/worlds/track_4.world
```

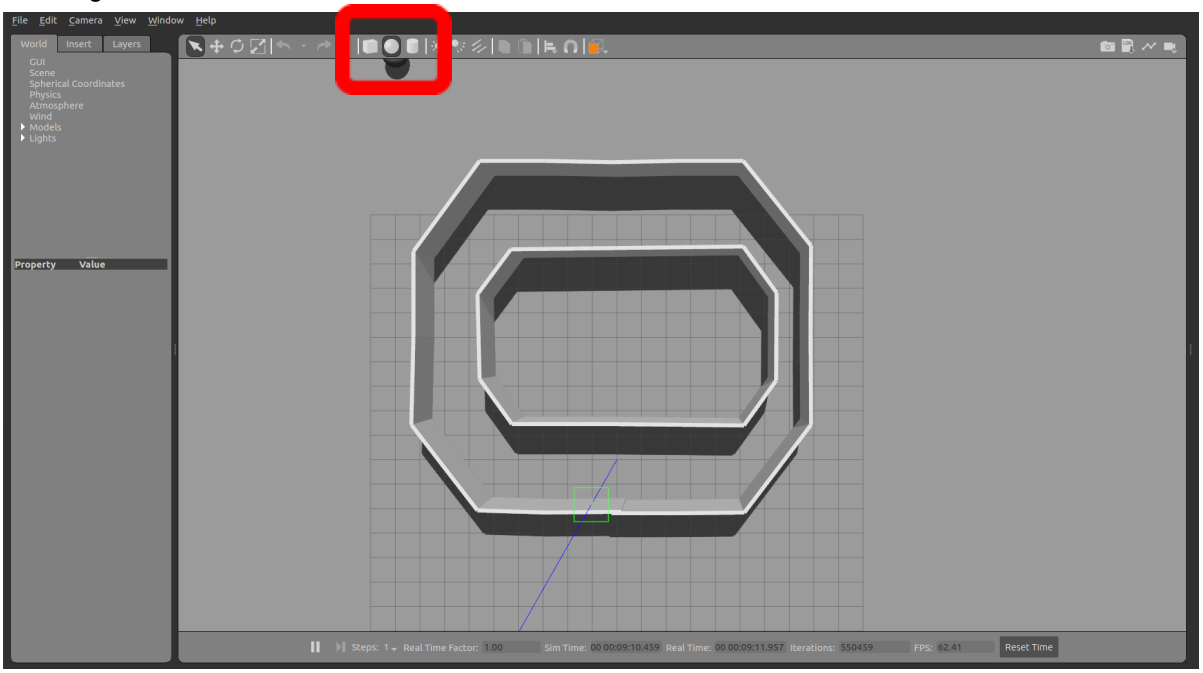


Now, without closing the gazebo window, make a new file named ‘track_{ID}_waypoints.txt’ in …/sim2real/worlds/waypoints folder with command below in another terminal window.

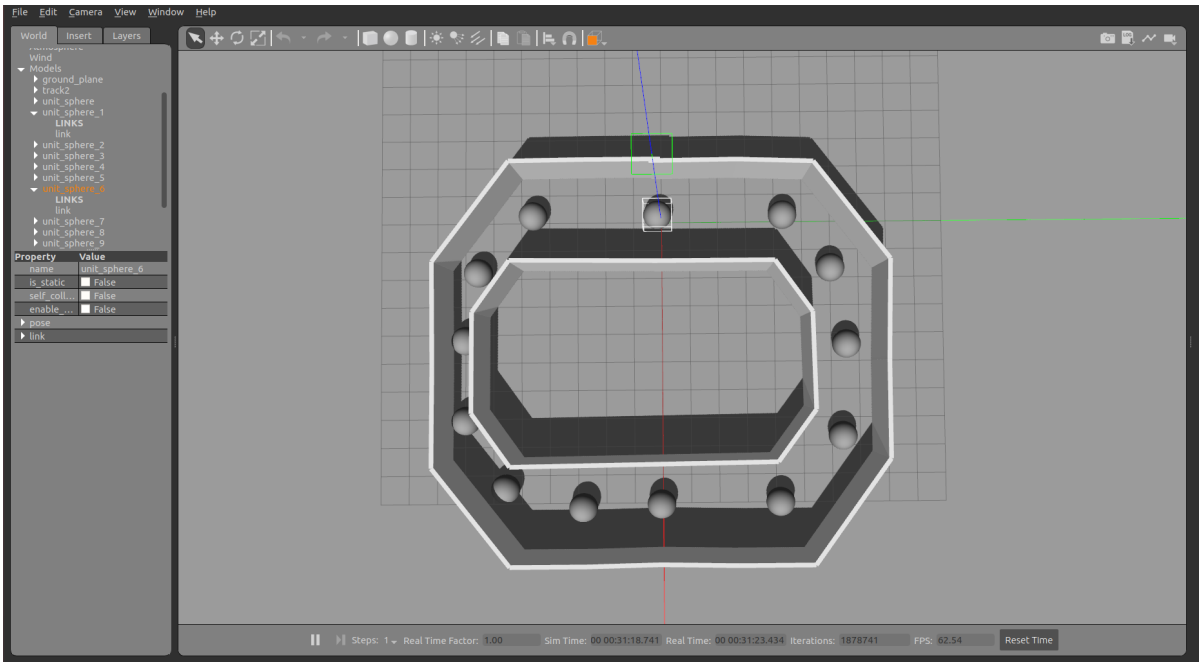
press ctrl + shift + t

```
gedit waypoints/track_{ID}_waypoints.txt
```

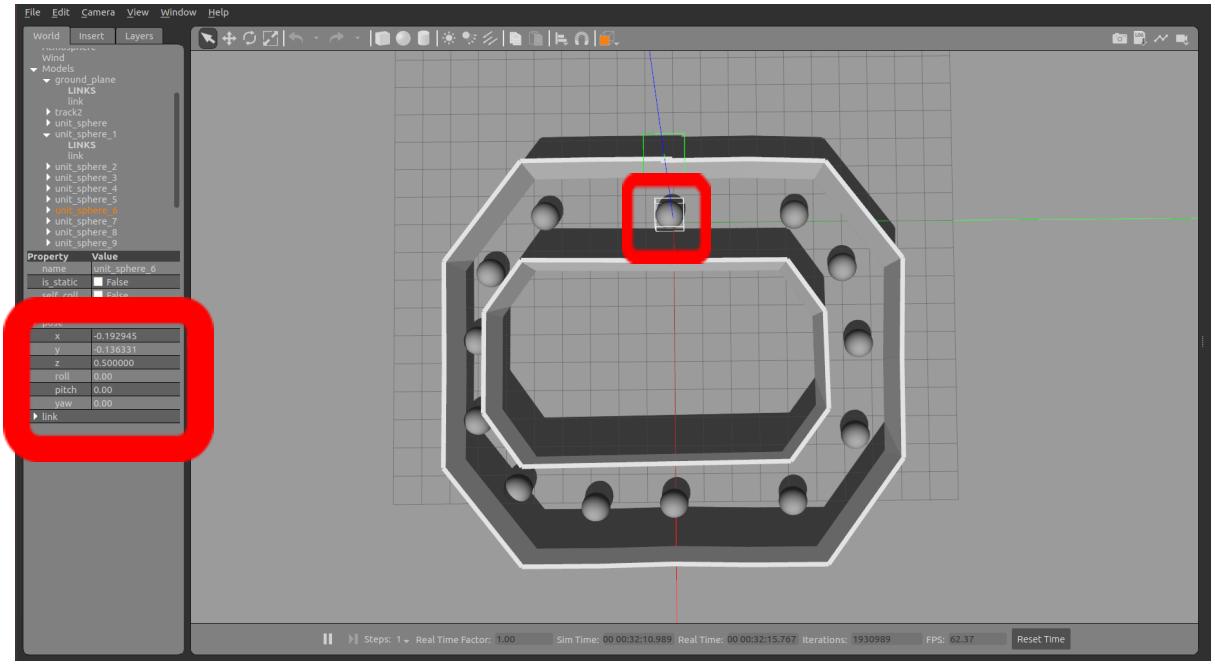
Now in gazebo window, select a ball in the above tools.



Place the balls in the positions where you think they are the critical points of all waypoints. These points will be automatically augmented

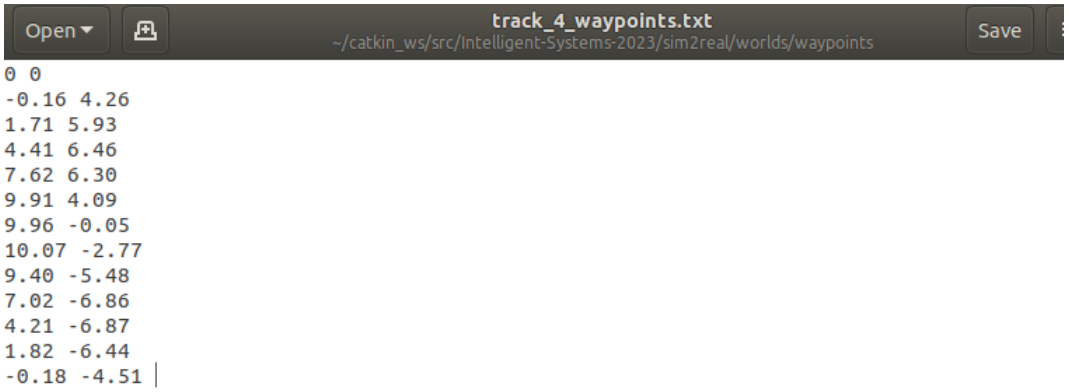


If you select the ball by clicking it, you can check the position of the ball on the left side of gazebo. The position values are in the ‘pose’ section.



(Caution - boring) Manually write down the x, y coordinates of all the balls to the txt file.

You should preserve the format which is shown in the already existing waypoint txt file. The first waypoint should be near the starting position(the intersection of red,green, blue lines).



Save and close the txt file and gazebo window. Move to …/sim2real/scripts/map_generation folder. Check where you are.

```
pwd
```

The output should be
/home/{USERNAME}/catkin_ws/src/Intelligent-Systems-2023/sim2real/scripts/map_generation

Interpolate the waypoints you have wrote down in the previous section with command:

```
python waypoint_augmentation.py --track_id {ID}

ex) python waypoint_augmentation.py --track_id 4
```

You can adjust the number of interpolations between two critical waypoints with the additional argument ‘divide’. The default value is 10.

```
ex) python waypoint_augmentation.py --track_id 4 --divide 20
```

Now the file with name ‘track_{ID}_aug_waypoints.txt’ is generated in …/sim2real/worlds/waypoints_augmented folder.

2.4. Generating world with waypoints

Make the world file with waypoints with command:

```
python waypoints_visualized_map.py --track_id {ID}

ex) python waypoints_visualized_map.py –track_id 4
```

If you follow all the previous steps well, you can see the coordinates of all the waypoints as an output of the command and can find the world file named ‘track_{ID}_waypoints.world’ in …/sim2real/worlds/worlds_with_waypoints folder.

2.5. Running codes in a new map

Remember what you want is to run the code you wrote in the previous project(Project 1).

Move to …/sim2real/scripts folder and check where you are.

```
pwd
```

The output should be /home/{USERNAME}/catkin_ws/src/Intelligent-Systems-2023/sim2real/scripts

Open eval_generator.py with gedit.

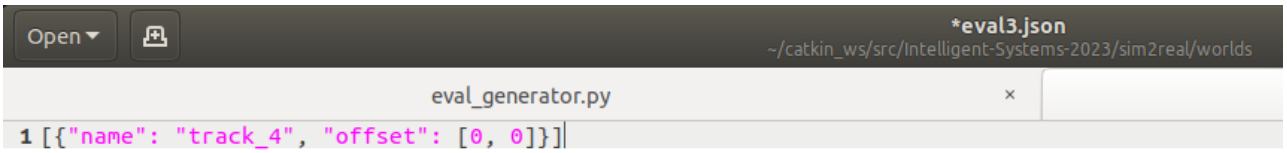
gedit eval_generator.py

and fix the values of ‘world_list’ and ‘offset’ variables in line 73 and 78 of eval_generator.py.

For example, in this tutorial, we will use track_4 we have made before and set the offset as [0,0].

```
72 def export_world_file(save_path):
73   world_list = [4] # Change this for your own custom worlds!
74   A = []
75   B = []
76   N_total = 0
77
78   offset = [[0, 0]] # Change this for your own worlds! You also need to write this information to sim2real/worlds/eval3.json
```

After that, you should also change the codes in eval3.json in …/sim2real/worlds folder to match with this information.



Make sure that you are now in …/sim2real/scripts and run the command below

```
python eval_generator.py
```

Finally, change the code in line 12 in {TEAMNAME}_project1.py in sim2real/project/{TEAMNAME}/project/ folder

from ‘eval1.yaml’ -> ‘eval3.yaml’

```
1 #!/usr/bin/env python2
2 from __future__ import print_function
3 import sys
4 import os
5 import rospkg
6 import rospy
7
8 ######################################## PLEASE CHANGE TEAM NAME ########################################
9 TEAM_NAME = "RLLAB"
10 ######################################## PLEASE CHANGE TEAM NAME ########################################
11 project_path = rospkg.RosPack().get_path("sim2real")
12 yaml_file = project_path + "/config/eval3.yaml"
13 PATH = project_path + "/scripts"
14 sys.path.append(PATH)
15
16 from sim2real.msg import Result, Query
17
18 import gym
19 import env
20 import numpy as np
21 import math
22 import yaml
23 import time
```

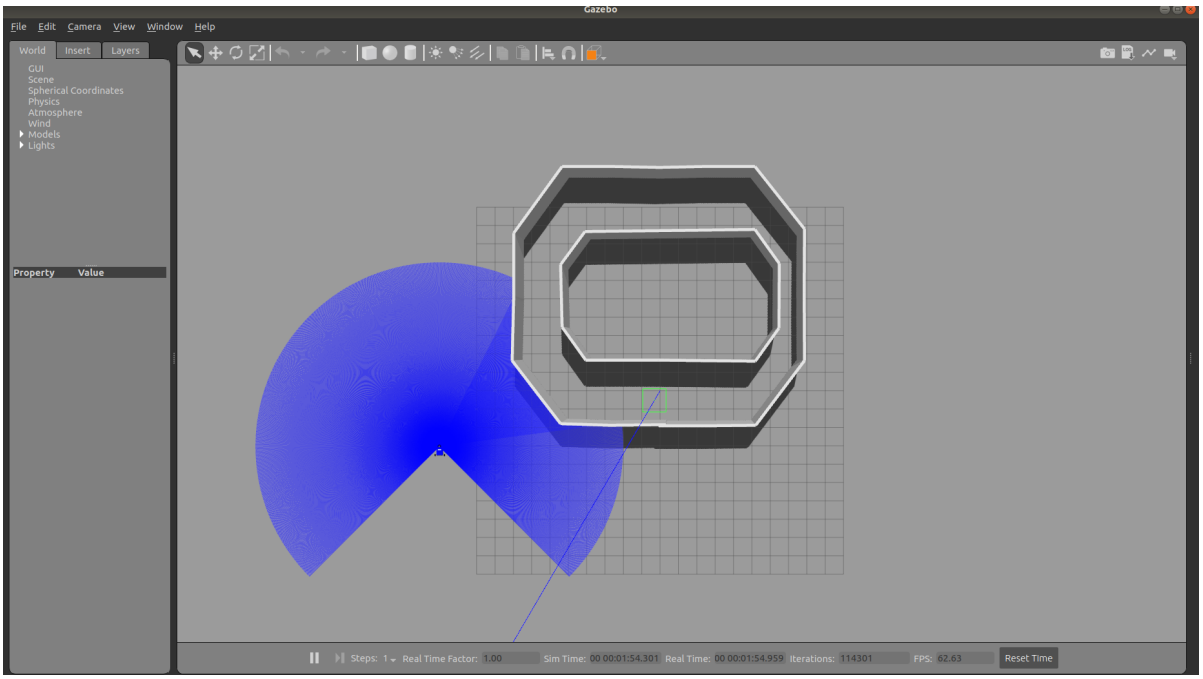
Repeat the process in project 1 but with base_p3.launch. Make sure that your virtual environment is activated in every terminal.

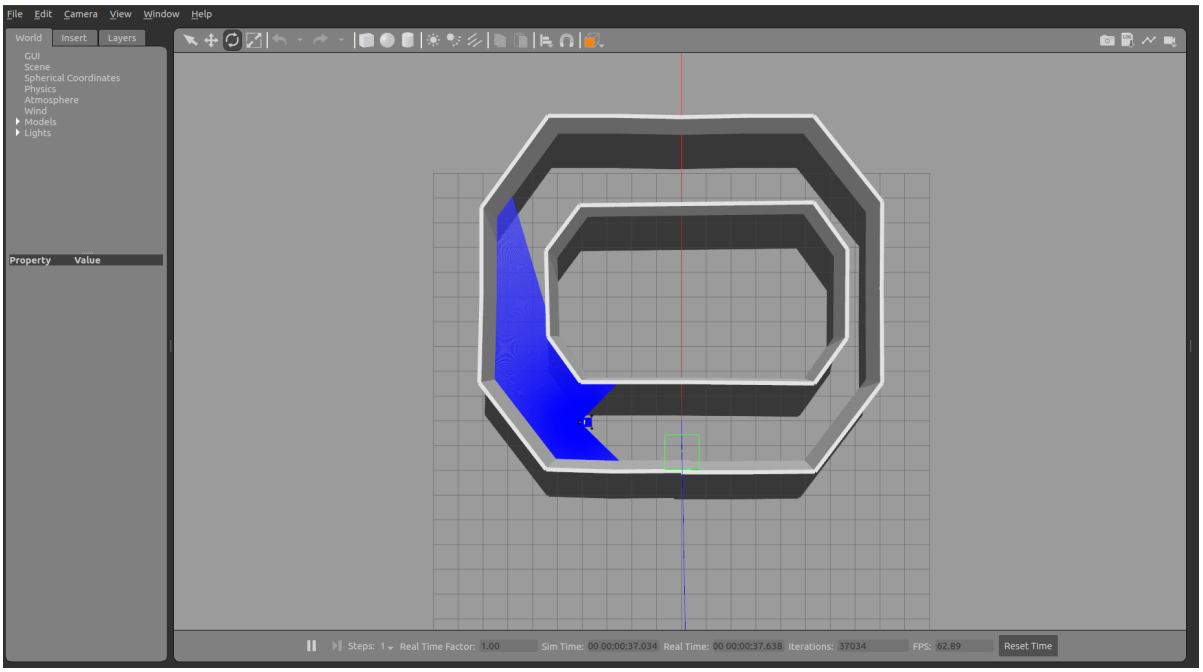
```
roslaunch sim2real base_p3.launch
```

```
rosrun sim2real {TEAM_NAME}_project1.py
```

```
rostopic pub /query sim2real/Query "{id: '0', trial: 0, name: '{TEAM_NAME}', world: 'track_1', exit: false}"
```

You can find your RC car running with the controller you made in project 1 in the new map .




File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property Value
Steps: 1 Real Time Factor: 1.00 Sim Time: 00 00:00:37.034 Real Time: 00 00:00:37.638 Iterations: 37034 FPS: 62.89
Reset Time