Gazebo Simulator

Unit 6: Final Project

- Summary -

Estimated time to completion: **8 hours**

Congratulations! If you are here, it means you have completed the course and are ready for the final project - an opportunity to apply everything you have learned so far!

- End of Summary -

- Final Project -

Summary

For the final project, you will prepare an entire simulation. Start from the ROS package, go through the world configuration with its models, connect it to Gazebo plugins, and create a robot connecting to ROS.

1. Preparing the environment

Create a new ROS package

- Create a new package called **final_project** that will contain everything related to this project
- Remember to add the dependencies you have used so far

2. Create a new world

Prepare Launch and World files

- Create a new **world** file and a **launch** file to start it using **roslaunch**

Populate this world

1. Create a new custom model called **proj_model**. Use the basic Gazebo shapes to create it.
2. Use walls (from the Gazebo library) to create a fence in the world. Make it a square with 20x20, so a robot will have space to work inside. (This is the default visible part of the ground plane).
3. Inside the fence, populate it with **four objects** of the custom model created, uniformly distributed in an **area of 10x10**.

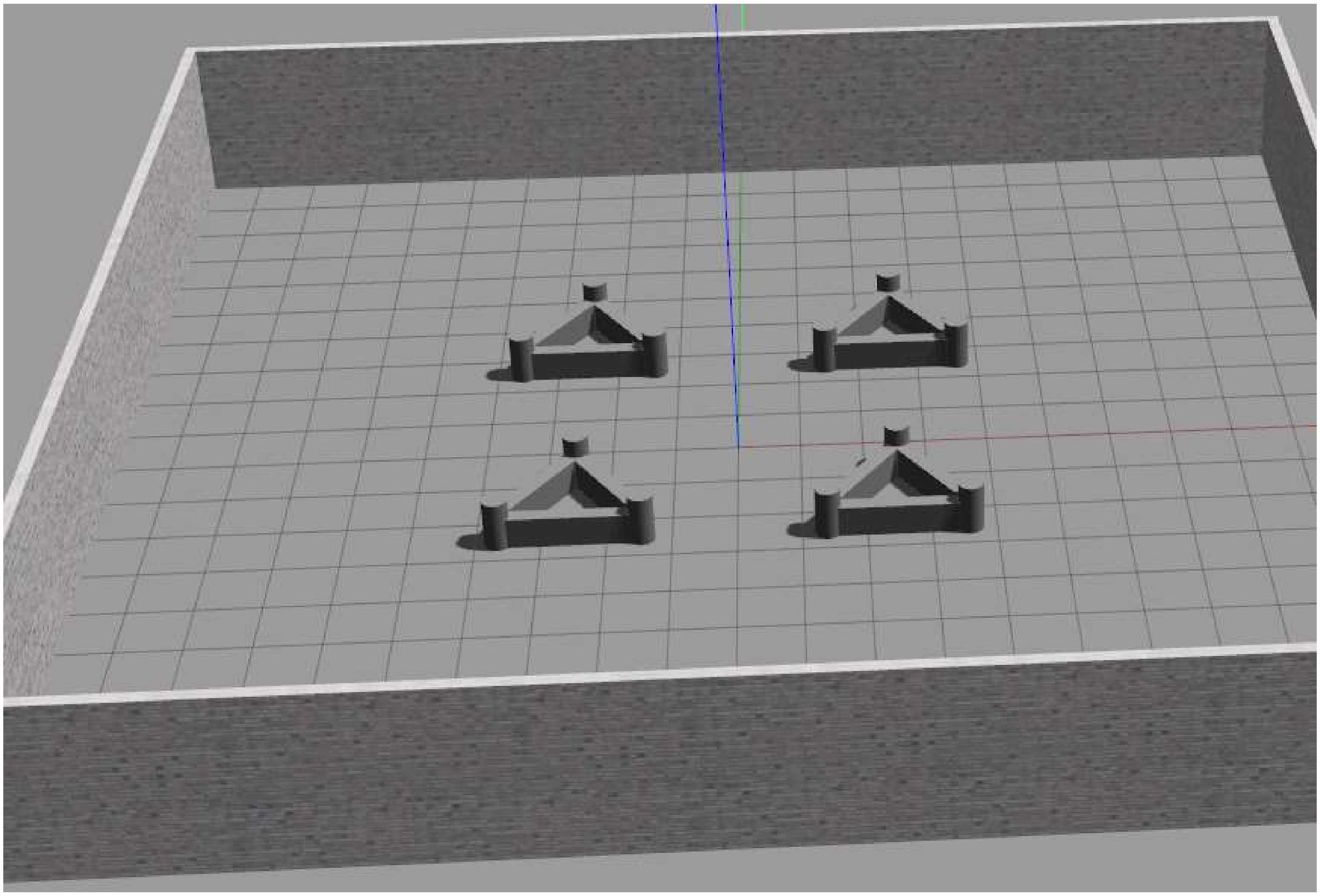
Hints:

- Create the world with the walls using Gazebo GUI
- Save the world file in your package
- Append the population of custom objects to the end of the world file

---

Example

The world created by you does not need to be exactly like that. This is just an example.



3. Write a model plugin

Prepare Files

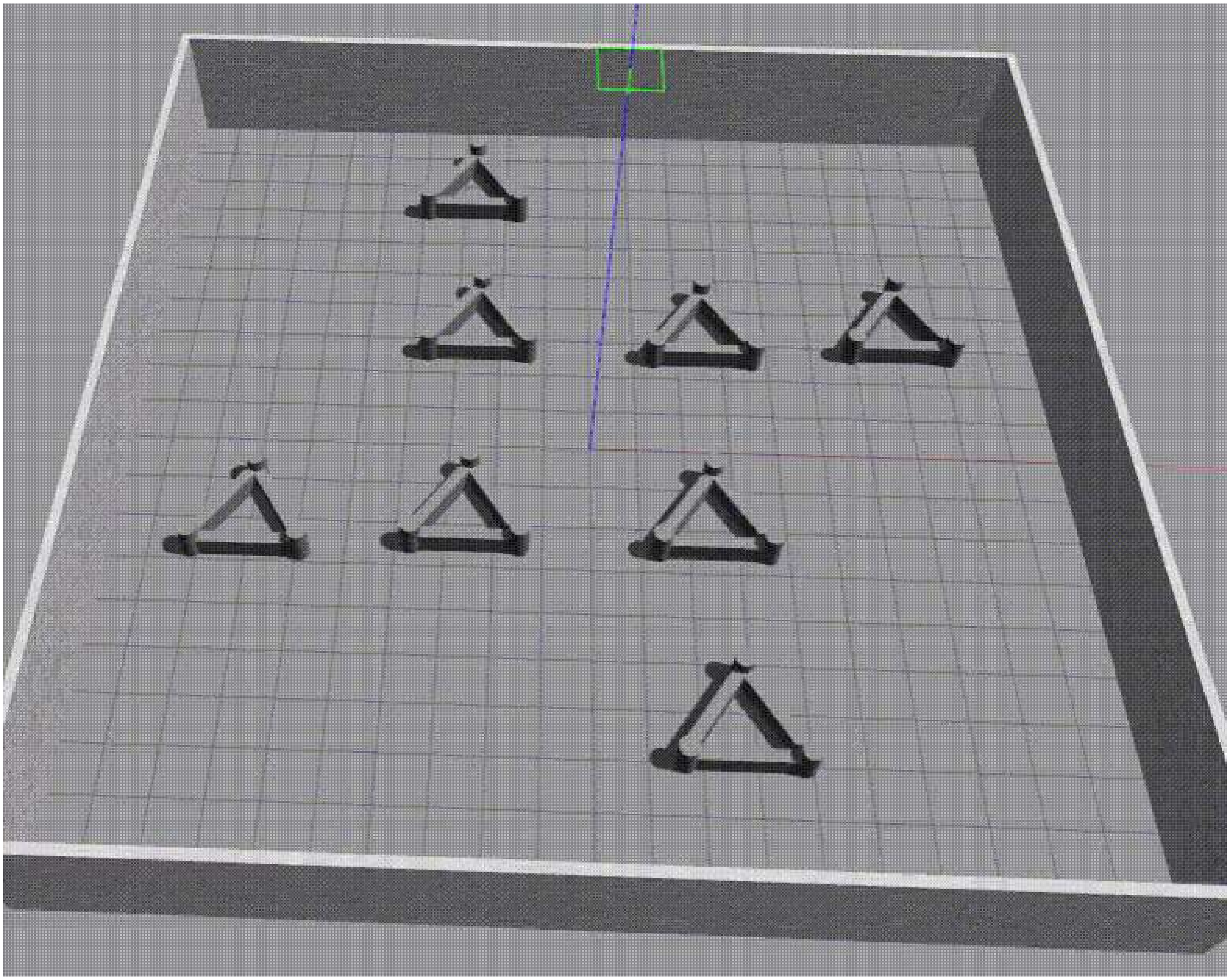
- Create your plugin file
- Prepare your **CMakeLists.txt** file to compile the new plugin properly.

Plugin description

1. This plugin must keep moving an object in a line (forwards and backward), so you create a kind of dynamic obstacle.
2. Put **four objects of your choice** with the plugin configured so the robot will also have to avoid this kind of obstacle.

Example

The world created by you does not need to be exactly like that. This is just an example. Although, the behavior of the model plugin must be similar.



4. Create a new robot

Prepare Files

- Create a **URDF** folder to contain robot files
- Use your knowledge about XACROs to help you create the new robot

Robot attributes

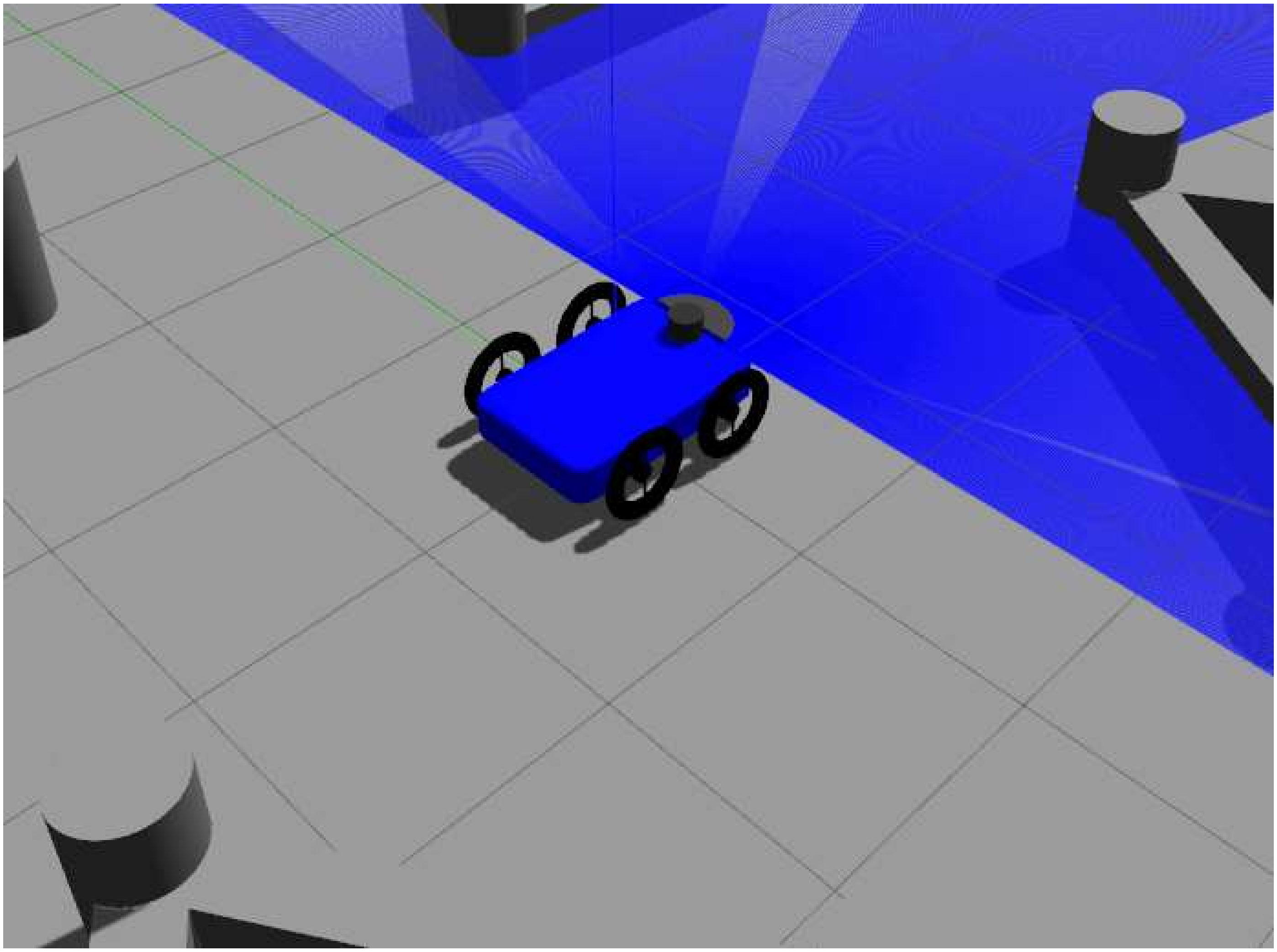
1. This new robot must have **four wheels**
2. Create a smaller robot this time. You want to test obstacle avoidance because it will help you to have it working better.
3. Use the plugin **skid steering drive** to control the robot and read its odometry data
4. Place a **laser scan** on top of the robot to avoid obstacles

Hints

1. You can use the same wheels from the robot created during the course
2. Create a different shape for the chassis and use a geometry tag from URDF. You do not have to have a custom DAE or STL file.
3. For skid-steer robots, the wheels must have a slippery factor. Remember to decrease a little bit the friction constants of the wheels.

Example

The robot created is just an example. You can have one slightly different. Check how much smaller it is than the previous one created. You are using four wheels with the skid steer driver.



5. Improve the obstacle avoidance algorithm and test it!

Prepare Files

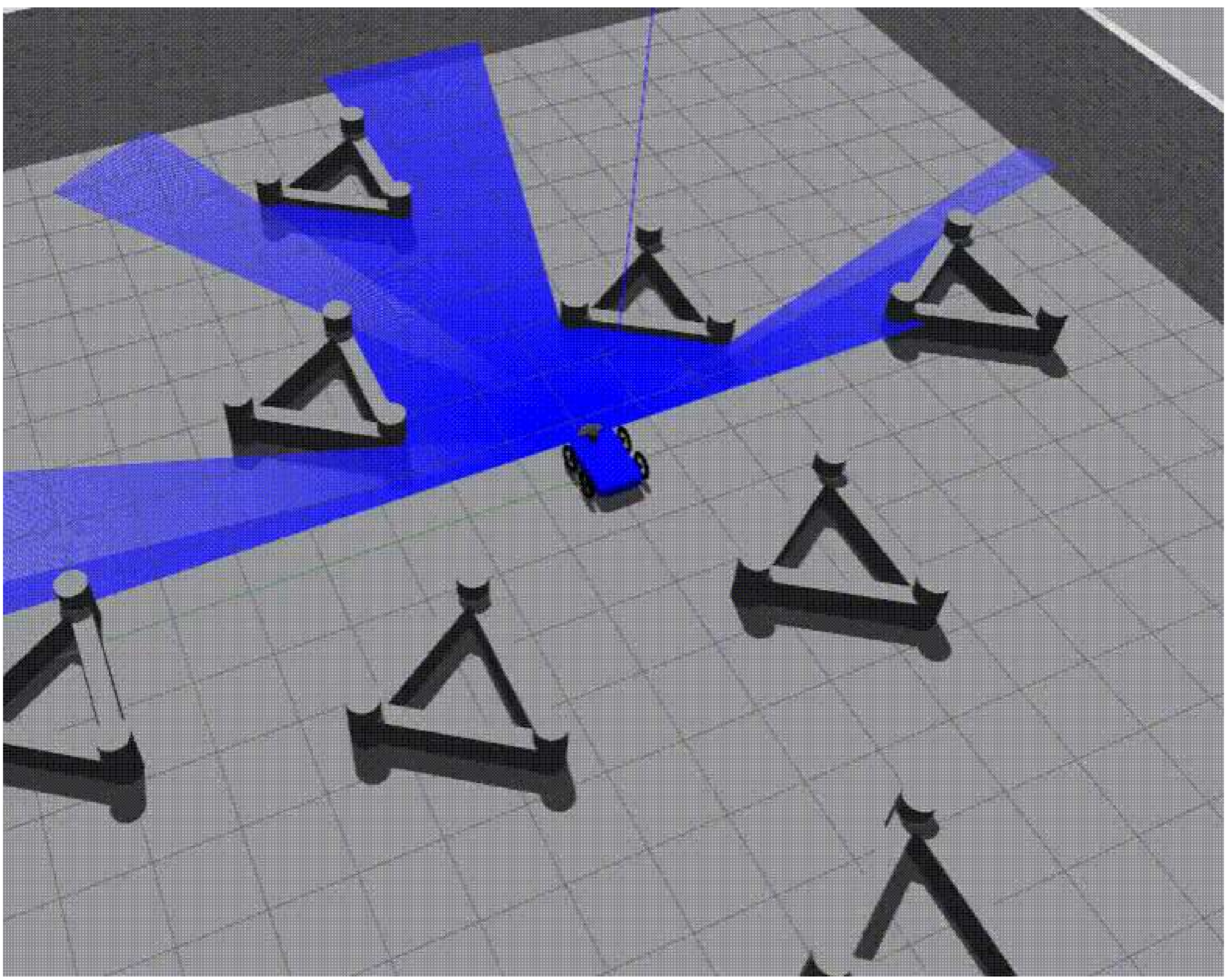
- Start from the given script file.

script.py (unit-06/obstacle_avoidance.py.txt)

- Improve obstacle avoidance. For example, the robot must go backward sometimes, because the obstacles may approach it in some cases since they are not static!

Expected behavior

1. The algorithm should be able to read your laser scan topic. It will print the mean values whenever they are updated in the callback of the script.
2. The robot must move around, going forward whenever it has free space and turning to the left side when an obstacle is close enough to the sensor.



- End Final Project -

- Solution -

Do not check the solution before trying to complete the exercise on your own! Instead, review the solution when you have finished the exercise. It is an excellent resource to compare with your solution.

If you have problems solving the exercise, use our forum support!

[final-project.zip (unit-06/final_project.zip)](unit-06/final_project.zip)

- End of Solution -