Cameron Ray
Phillip Voss
Yilei Shen
Zachary Kasper

World and launch documentation

The world file describing the layout of the world is found in the launch folder of proj1 and is named project1.world. The world name is set to default and the sun name is set to the sun. It is set to a be coming down at an angle so as to cast shadows by the object in our world as well as the robot itself. We are using all of the default light setting for gazebo when using the sun as the light source. The next object created in the file is the ground. Its name is ground_plane and is also using the default Gazebo ground object including the collison fucntion and friction function. The visual settings are also set to default. This includes default shadow and material settings.

Next is the gravity function which once again is using the default settings and the pull of gravity is set to -9.8. Magnetic fields, atmospheric settings, and the world's physics are set to default values. The scene is then set with ambient light settings as well as the background and shadows. These parameters are all set to the default values. Following the scene settings, the spherical coordinatates of the world are set to the earth's actual surface model in addition to choosing a latitude, longitude, elevation, and heading all of 0. The next object defined is the state of the world. This includes the simulated time relative to real time and wall time. These are all set to default times. The ground velocity is then set to 0, the acceleration is set to 0, and the wrench is set to 0.

The following objects in the world file is the wall which we drew into the world depends on the requirement. We converted the feet into meter because the gazebo only uses meter as its default unit, but we still use feet to name the wall, it is more clear and uniform related with requirement. We drew the walls with three different length which is 5 feet, 10 feet and 20 feet, then clone them severial times for different properties. Used the walls to form the map which is used for robot testing.

The XML launch file is found in the launch folder of proj1 and is named proj1.launch. We can modified this file to change the setting of launch. After we modified it, we use "catkin_make" to update, then use ".~/catkin_ws/devel/setup.bash" to let us have the ability to open the launch file. Then, use the "roslaunch proj1 proj1.launch" to open the launch file

and run our application. All the contents is between the pair of <launch> and </launch>. The first argument is the name and the working directory of our world file. It is the place we need to modify during developing the application individually. Second part is the additional parameters taken from turtlebot_world.launch. We can modify this part to set up the robot status. We also added the rviz node to make users open the rviz function when running the launch file. The third part of the keyboard section to modify the linear and the angular. And we added our main code node into this section called proj1 using the type of project_node.py which is the node name of our application file. The last section is the laser section which is used to manipulate the laser of robot like changing the scan hight and changing the value of laser range.

Instrtructions

 # First run these to setup the file structure:

```
mkdir -p ~/catkin_ws/src
cd ~/catkin_ws/src
catkin_make
catkin_create_pkg proj1 std_msgs rospy roscpp
cd ..
catkin_make
. ~/catkin_ws/devel/setup.bash
```

 # Then replace the proj1 folder with the proj1 canvas folder and then rebuild:

```
cd ~/catkin_ws
catkin_make
. ~/catkin_ws/devel/setup.bash
```

 # Then nodes go into proj1/src, and need to be added to the proj1.launch file, run the launch file:

```
roslaunch proj1 proj1.launch
```