# Creating World Environment

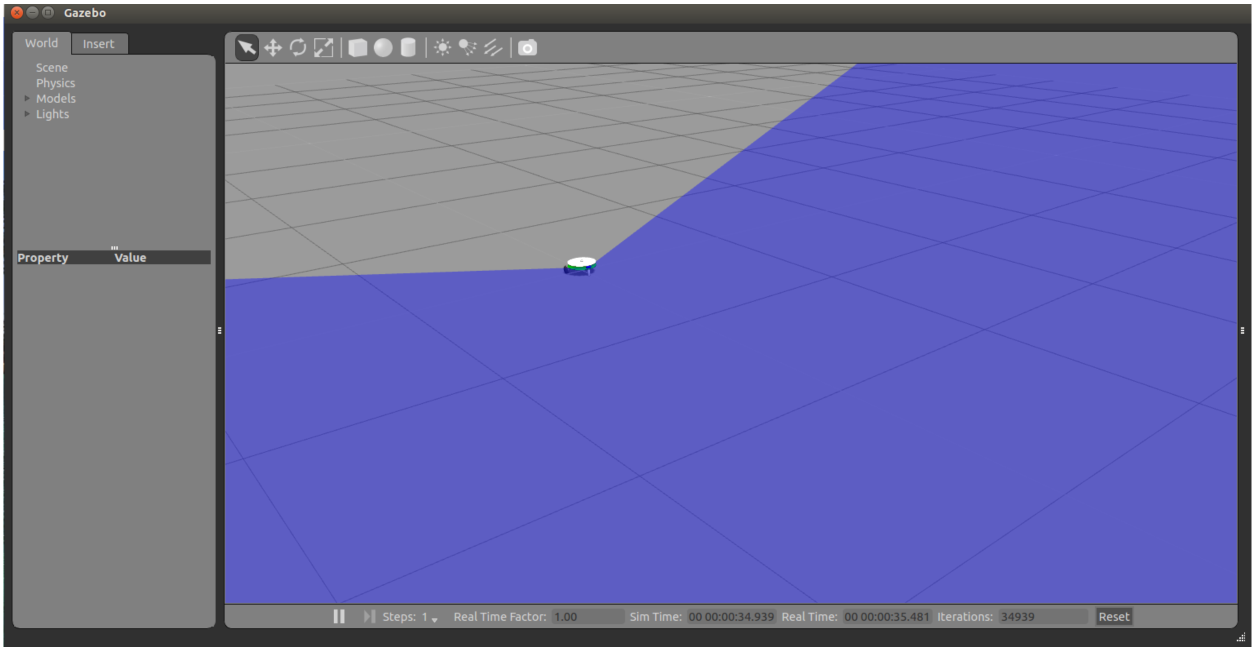
In the previous tutorial, you have learnt how to launch and move the robot in gazebo and RViz. In this tutorial, you will learn how to choose the world for simulation **and create a new world**.

Follow the instructions given below to create a new world:

1. To launch an **empty world** in gazebo, you can simply run the following command in terminal:

   ***roslaunch task_1 simple_robot_gazebo.launch***

   This command will launch an **empty world with robot model** as shown in Figure 1.1



Steps: 1 Real Time Factor: 1.00 Sim Time: 00 00:00:34.939 Real Time: 00 00:00:35.481 Iterations: 34939

Figure 1.1: Empty world

2. You can add the simple object and shape using the insert tab
   - Click on **insert tab**
   - Choose the object in the given list to create the world

   Refer to Figure 1.2 for a list of tools in gazebo.

**NOTE:** You will see **connecting to model database**. It might take a long time to connect to Gazebo's model database.

3. Now you can **add** the **different objects from the list to environment**:
   - Add a "gray wall"
   - Add a "dumpster"

- Add a “square box”
- Add a “bookshelf” to create the world

If we add the above objects, after editing world will look like Figure 1.3.

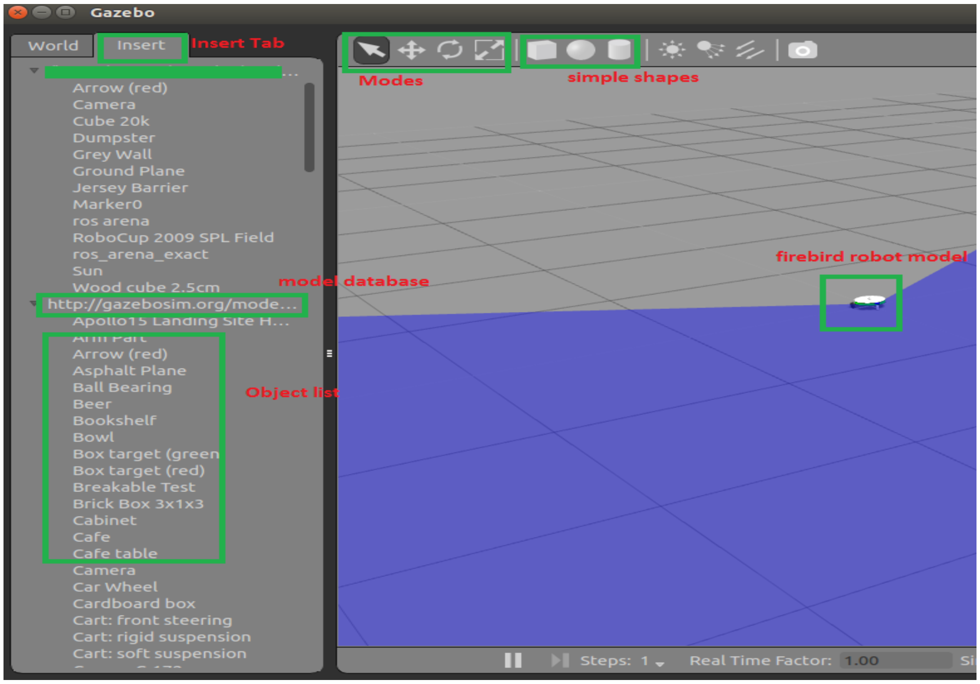
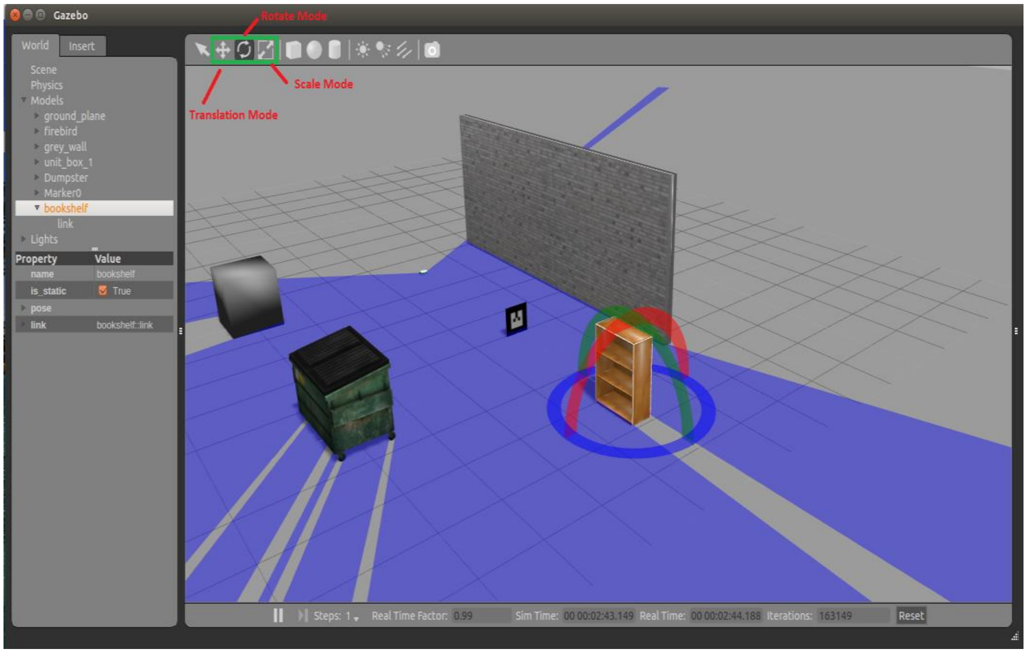


Figure 1.2 Tools of gazebo simulator



| Property | Value |
|---|---|
| name | bookshelf |
| is_static | ☑ True |
| pose | |
| link | bookshelf::link |

Figure 1.3: Creating World

You can use the different modes as shown in Figure 1.3 to play with these objects.

4. **Save** the world
    - Press **Shift + Ctrl + S** to save the world. It will open a new window then navigate to the path where you want to store this world file. Please save this file **in your package** inside a **"world"** directory.
    - Make sure that your file type is **".world"**. Example: testing.world

Verify that your file is saved by visiting the world directory.

5. Launch gazebo with the world
    - Navigate to launch folder of your package using command:

      ***roscd task_1/launch***
    - Open **simple_robot_gazebo.launch** in text editor using following command:

      ***gedit simple_robot_gazebo.launch***

    You will find the following content in the launch file:

```
<launch>
<!-- these are the arguments you can pass this launch file, for
example, paused:=true -->

    <arg name="paused" default="false"/>
    <arg name="use_sim_time" default="true"/>
    <arg name="gui" default="true"/>
    <arg name="headless" default="false"/>
    <arg name="debug" default="false"/>

<!-- We resume the logic in empty_world.launch -->
<include file="$(find gazebo_ros)/launch/empty_world.launch">
    <arg name="debug" value="$(arg debug)" />
    <arg name="gui" value="$(arg gui)" />
    <arg name="paused" value="$(arg paused)"/>
    <arg name="use_sim_time" value="$(arg use_sim_time)"/>
    <arg name="headless" value="$(arg headless)"/>
<!—
<arg name="world_name" value="$(find task_1)/world/arena_testing.world"/>
-->
</include>
<!-- Load the URDF into the ROS Parameter Server -->

    <param name="robot_description" command="$(find xacro)/xacro.py '$(find
task_1)/urdf/firebird.urdf'" />
```

```
<!-- Run a python script to the send a service call to gazebo_ros to spawn a URDF
robot -->

    <node name="urdf_spawner" pkg="gazebo_ros" type="spawn_model"
respawn="false" output="screen"
    args="-urdf -model firebird -param robot_description"/>

</launch>
```

- Find the **highlighted lines** shown above in your launch file
- Firstly **uncomment the line** and **change the path of world file** in this launch file to launch **your world** in the gazebo environment.

6. Now save the launch file and run the command in terminal to launch the **simple_robot_gazebo.launch** file.

You can find the full tutorial about creating/editing the Gazebo world here.