PRESENTS

ARTIFICIAL INTELLIGENCE in MOBILITY

ARTIFICIAL INTELLIGENCE IN MOBILITY

GAZEBO SIMULATOR

IN COLLABORATION WITH

Time Of Sports®

COMPANY CONFIDENTIAL

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

WHAT IS GAZEBO?

Gazebo is a 3D dynamic simulator with the ability to accurately and efficiently simulate populations of robots in complex indoor and outdoor environments.

Typical uses of Gazebo include:

- testing robotics algorithms,
- designing robots,
- performing regression testing with realistic scenarios

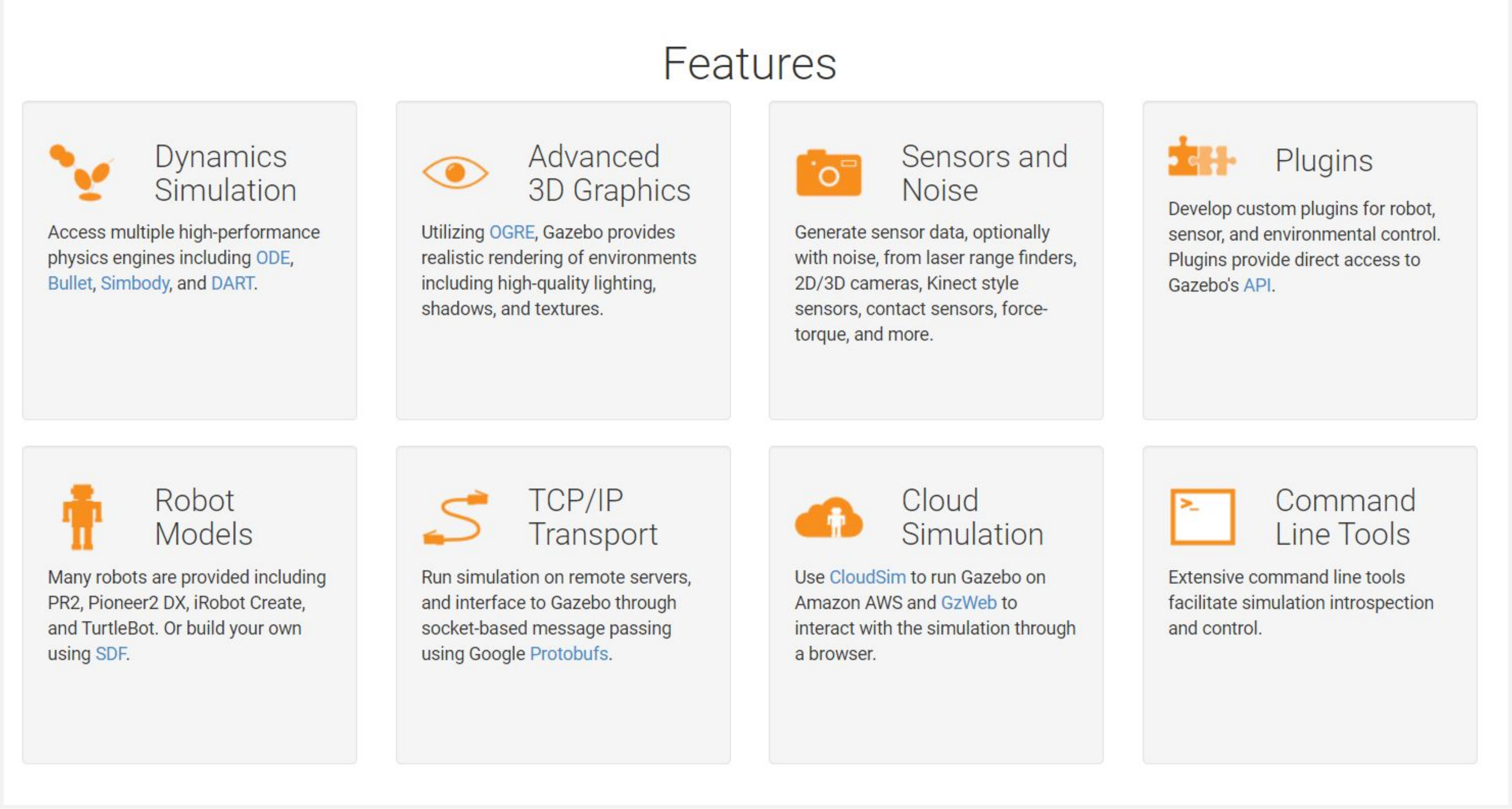
A few key features of Gazebo include:

- multiple physics engines,
- a rich library of robot models and environments,
- a wide variety of sensors,
- convenient programmatic and graphical interfaces

GAZEBO

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

COMPANY CONFIDENTIAL



Features

Dynamics Simulation

Access multiple high-performance physics engines including ODE, Bullet, Simbody, and DART.

Advanced 3D Graphics

Utilizing OGRE, Gazebo provides realistic rendering of environments including high-quality lighting, shadows, and textures.

Sensors and Noise

Generate sensor data, optionally with noise, from laser range finders, 2D/3D cameras, Kinect style sensors, contact sensors, force-torque, and more.

Plugins

Develop custom plugins for robot, sensor, and environmental control. Plugins provide direct access to Gazebo's API.

Robot Models

Many robots are provided including PR2, Pioneer2 DX, iRobot Create, and TurtleBot. Or build your own using SDF.

TCP/IP Transport

Run simulation on remote servers, and interface to Gazebo through socket-based message passing using Google Protobufs.

Cloud Simulation

Use CloudSim to run Gazebo on Amazon AWS and GzWeb to interact with the simulation through a browser.

Command Line Tools

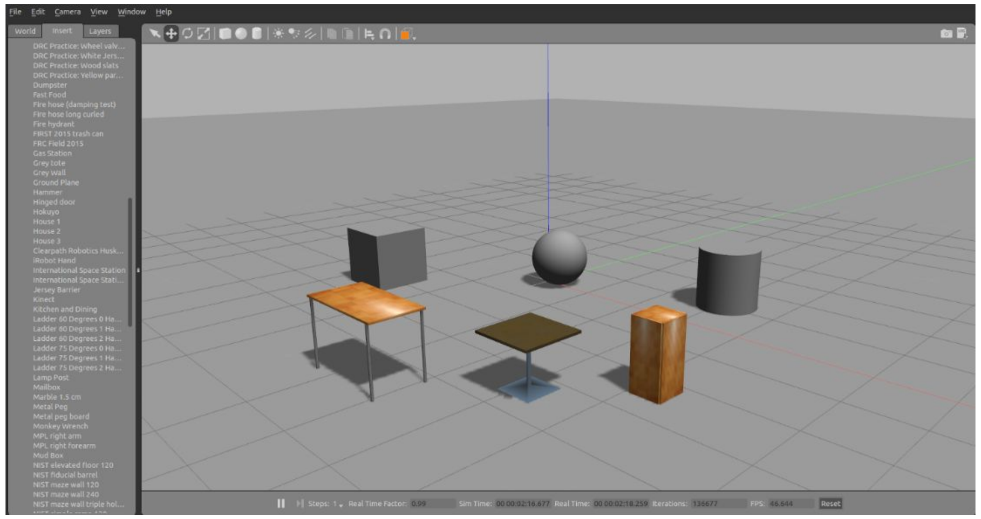
Extensive command line tools facilitate simulation introspection and control.

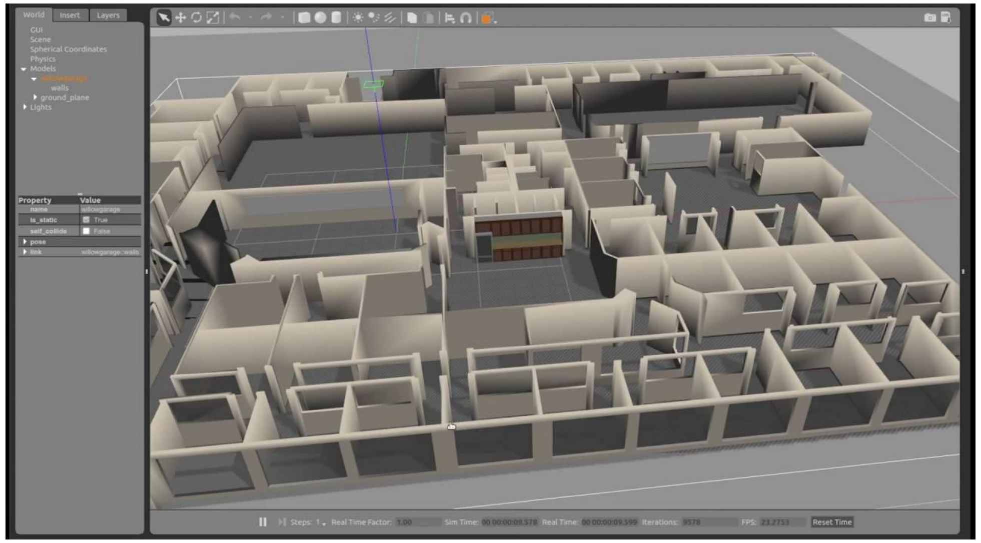
NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

World Files

The world description file contains all the elements in a simulation, including robots, lights, sensors, and static objects. This file is formatted using SDF (Simulation Description Format), and typically has a .world extension.

The Gazebo server (gzserver) reads this file to generate and populate a world.





NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

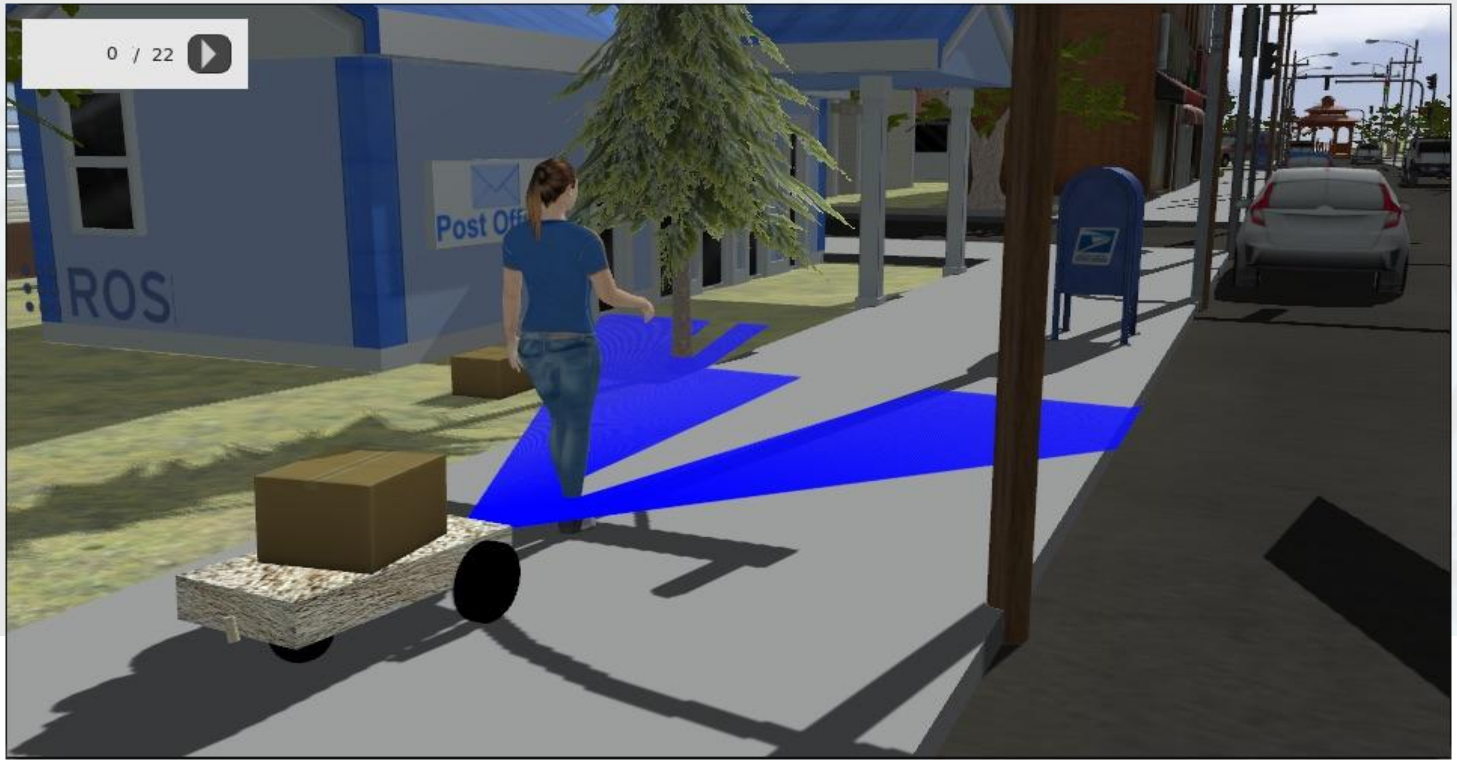
COMPANY CONFIDENTIAL

Terminology

•**World:** The term used to describe a collection of robots and objects (such as buildings, tables, and lights), and global parameters including the sky, ambient light, and physics properties.

•**Static:** Entities marked as static (those having the <static>true</static> element in SDF), are objects which only have collision geometry. All objects which are not meant to move should be marked as static, which is a performance enhancement.

•**Dynamic:** Entities marked as dynamic (either missing the <static> element or setting false in SDF), are objects which have both inertia and a collision geometry.



NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

COMPANY CONFIDENTIAL

GAZEBO COMPONENTS

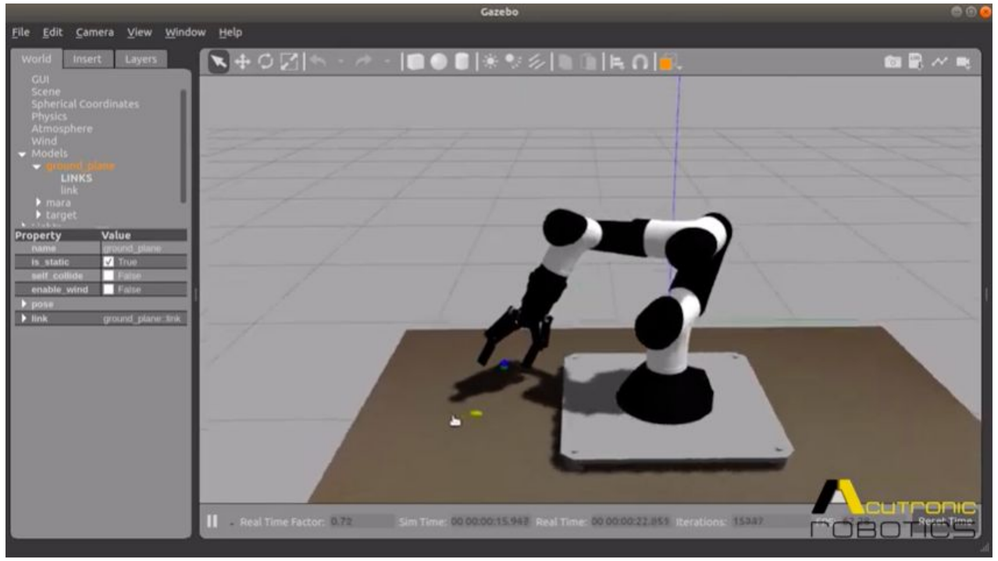
Model Files

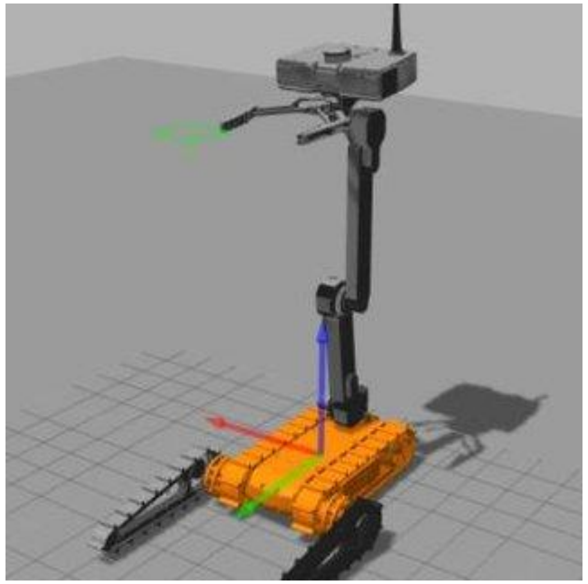
A model file uses the same SDF format as world files, but should only contain a single <model> ... </model>.

The purpose of these files is to facilitate model reuse, and simplify world files. Once a model file is created, it can be included in a world file using the following SDF syntax:

<include> <uri>model://model_file_name</uri> </include>

A number of models are provided in the online model database





NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

Components of SDF Models

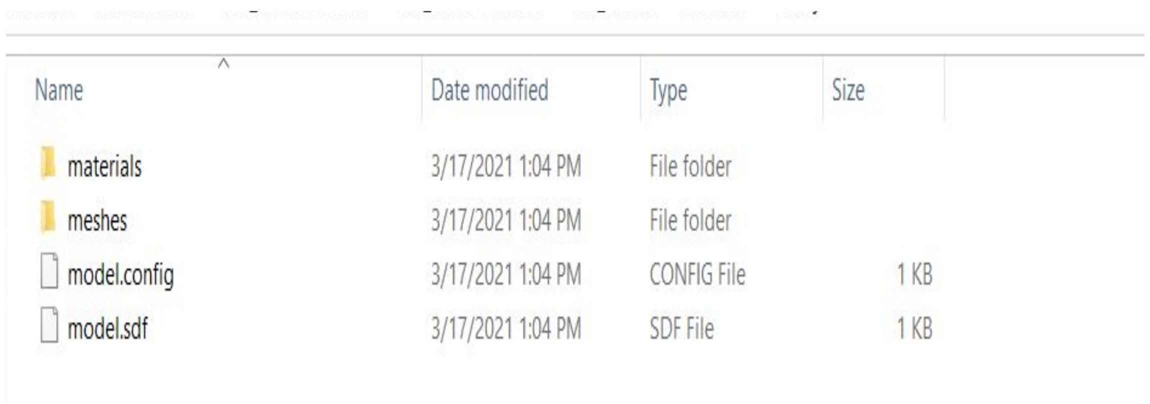
- **Links:** A link contains the physical properties of one body of the model. Each link may contain many collision and visual elements. Try to reduce the number of links in your models in order to improve performance and stability
- **Collision:** A collision element encapsulates a geometry that is used for collision checking. A link may contain many collision elements.
- **Visual:** A visual element is used to visualize parts of a link. A link may contain 0 or more visual elements.
- **Inertial:** The inertial element describes the dynamic properties of the link, such as mass and rotational inertia matrix.
- **Sensor:** A sensor collects data from the world for use in plugins. A link may contain 0 or more sensors.
- **Light:** A light element describes a light source attached to a link. A link may contain 0 or more lights.
- **Joints:** A joint connects two links. A parent and child relationship is established along with other parameters such as axis of rotation, and joint limits.
- **Plugins:** A plugin is a shared library created by a third party to control a model.

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

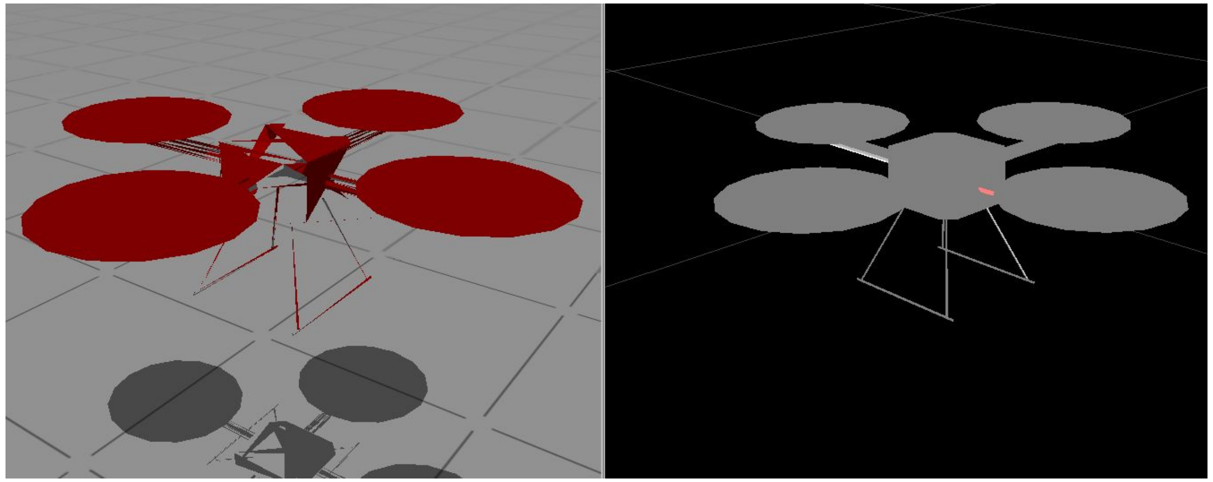
COMPANY CONFIDENTIAL

Model Database Structure

- *model.config* : Meta-data about model
- *model.sdf* : SDF description of the model
- *model.sdf.erb* : Ruby embedded SDF model description
- *meshes* : A directory for all COLLADA and STL files
- *materials* : A directory which should only contain the textures and scripts subdirectories
    - *textures* : A directory for image files (jpg, png, etc).
    - *scripts* : A directory for OGRE material scripts
- *plugins*: A directory for plugin source and header files



Name	Date modified	Type	Size
materials	3/17/2021 1:04 PM	File folder	
meshes	3/17/2021 1:04 PM	File folder	
model.config	3/17/2021 1:04 PM	CONFIG File	1 KB
model.sdf	3/17/2021 1:04 PM	SDF File	1 KB



Plugins

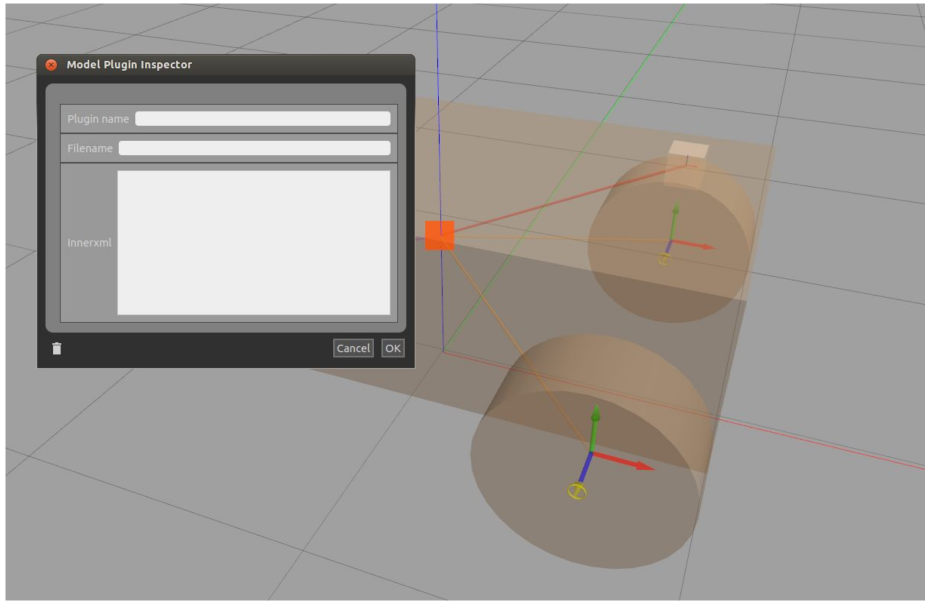
Plugins provide a simple and convenient mechanism to interface with Gazebo. A plugin is a chunk of code that is compiled as a shared library and inserted into the simulation. The plugin has direct access to all the functionality of Gazebo through the standard C++ classes.

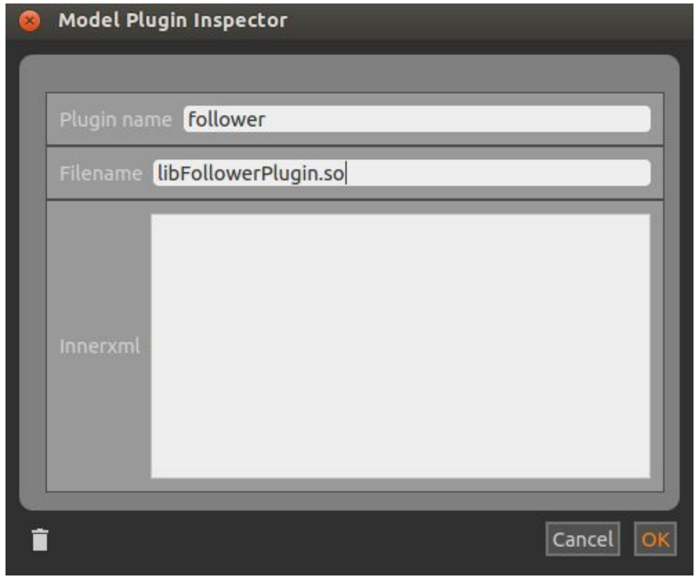
Plugins are useful because they:

- let developers control almost any aspect of Gazebo
- are self-contained routines that are easily shared
- can be inserted and removed from a running system

You should use a plugin when:

- you want to programmatically alter a simulation
- you want a fast interface to gazebo, without the overhead of the transport layer




Model Plugin Inspector

Plugin name: follower

Filename: libFollowerPlugin.so

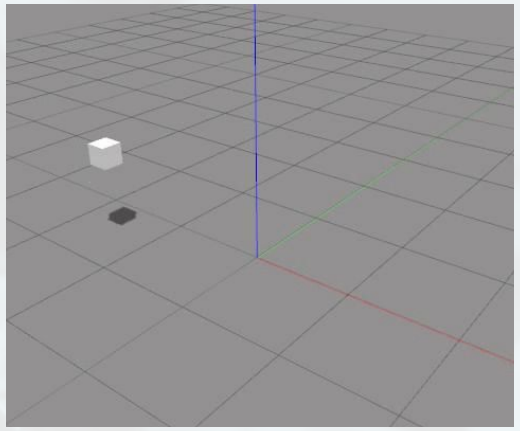
NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

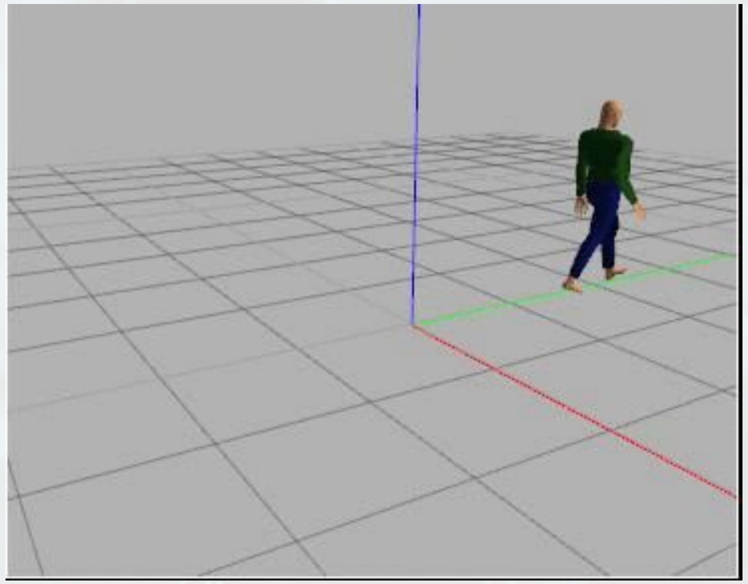
COMPANY CONFIDENTIAL

PLUGIN TYPES

There are currently 6 types of plugins

1. World
2. Model
3. Sensor
4. System
5. Visual
6. GUI





Each plugin type is managed by a different component of Gazebo. For example,

- a Model plugin is attached to and controls a specific model in Gazebo.
- a World plugin is attached to a world
- a Sensor plugin to a specific sensor.
- a System plugin is specified on the command line, and loads first during a Gazebo startup. This plugin gives the user control over the startup process
- a World plugin to control world properties, such as the physics engine, ambient lighting, etc

A plugin type should be chosen based on the desired functionality.

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

COMPANY CONFIDENTIAL

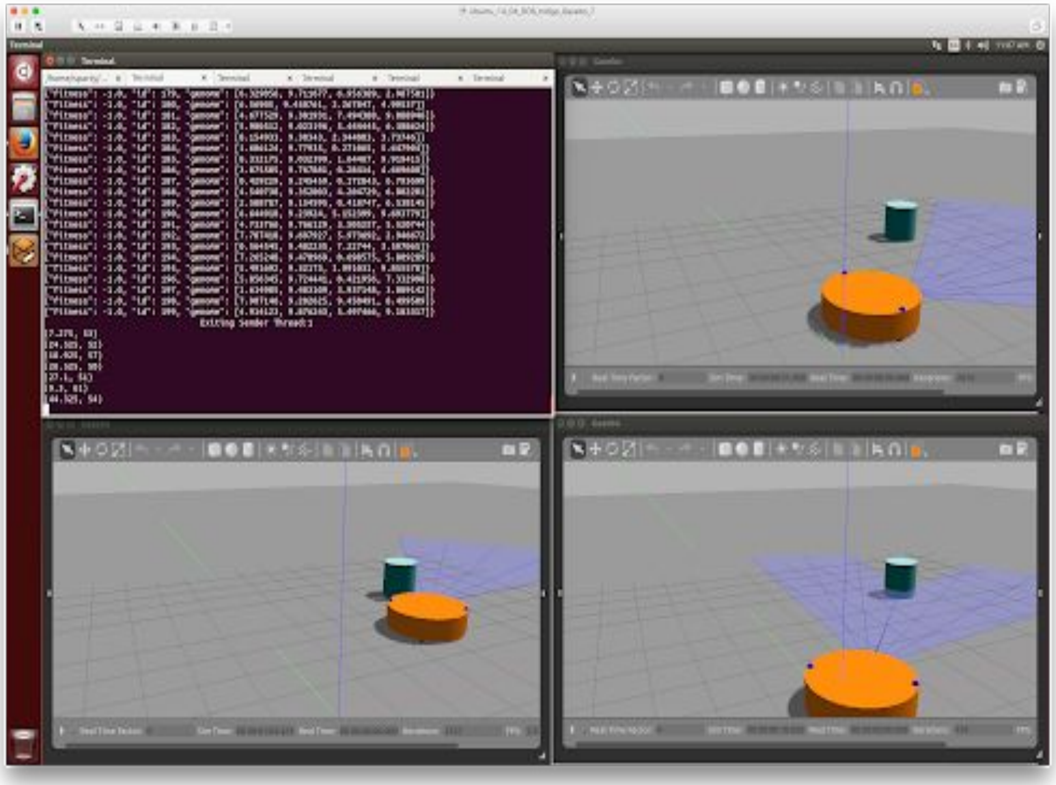
GAZEBO COMPONENTS

Gazebo Server

The server is the workhorse of Gazebo. It parses a world description file given on the command line, and then simulates the world using a physics and sensor engine.

The gzserver executable runs the physics update-loop and sensor data generation. This is the core of Gazebo, and can be used independently of a graphical interface. You may see the phrase "run headless" thrown about. This phrase equates to running only the gzserver.

An example use case would involve running gzserver on a cloud computer where a user interface is not needed.



```
[illegible]
                Exiting Sender Thread:
[illegible]
```

GAZEBO COMPONENTS

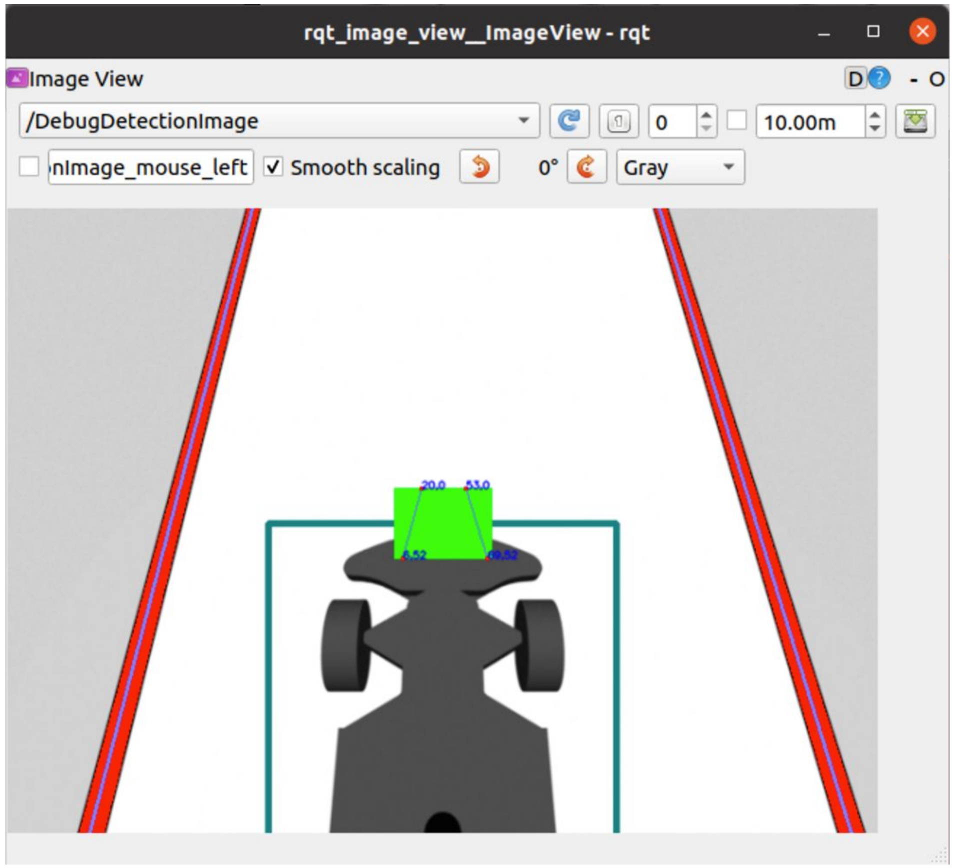
Graphical Client

The gzclient executable runs a QT based user interface.

The graphical client connects to a running gzserver and visualizes the elements. This is also a tool which allows you to modify the running simulation.

Server + Graphical Client in one

The gazebo command combines server and client in one executable. Instead of running gzserver and then gzclient, one can simply use gazebo command.



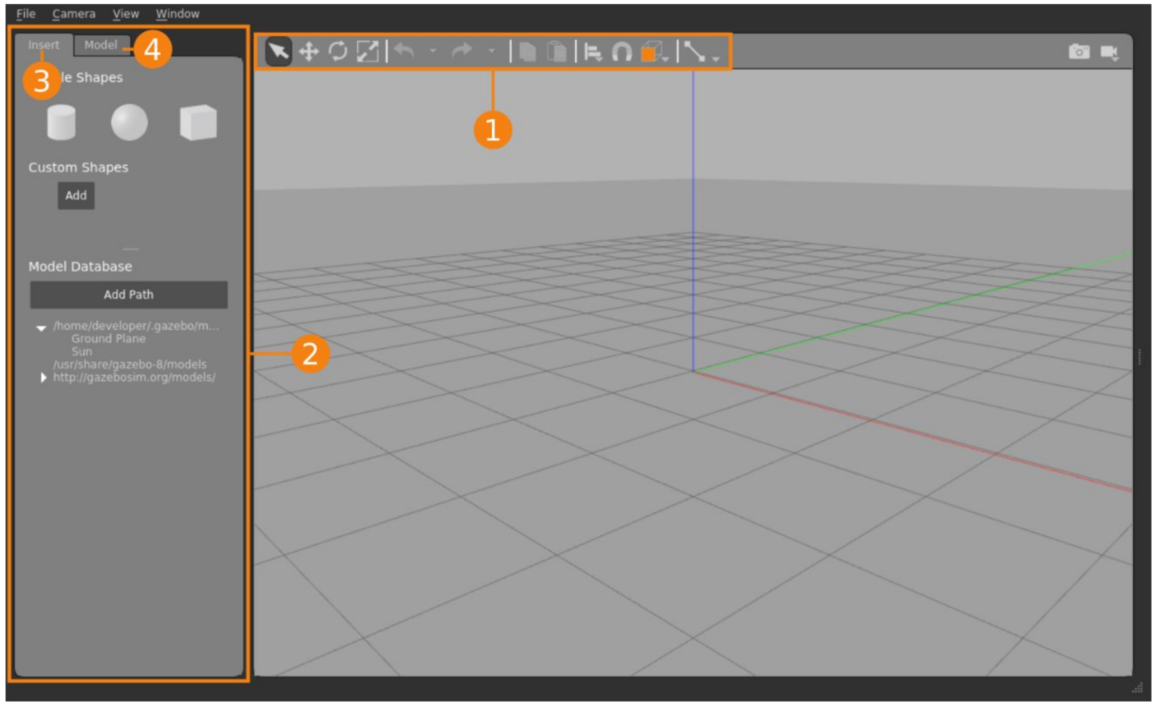
NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

Model Editor

The Model Editor lets us construct simple models right in the Graphical User Interface (GUI). For more complex models, you'll need to learn how to write SDF files, and check out the tutorials on building a robot.

Physics and the simulation will be paused as soon as you are in the Model Editor.

The Model Editor interface looks similar to the main Gazebo UI but with some subtle differences. The left panel and top Toolbar now contain only widgets for editing and creating parts of the model. The bottom Toolbar that displays simulation data is hidden since the simulation is now paused.



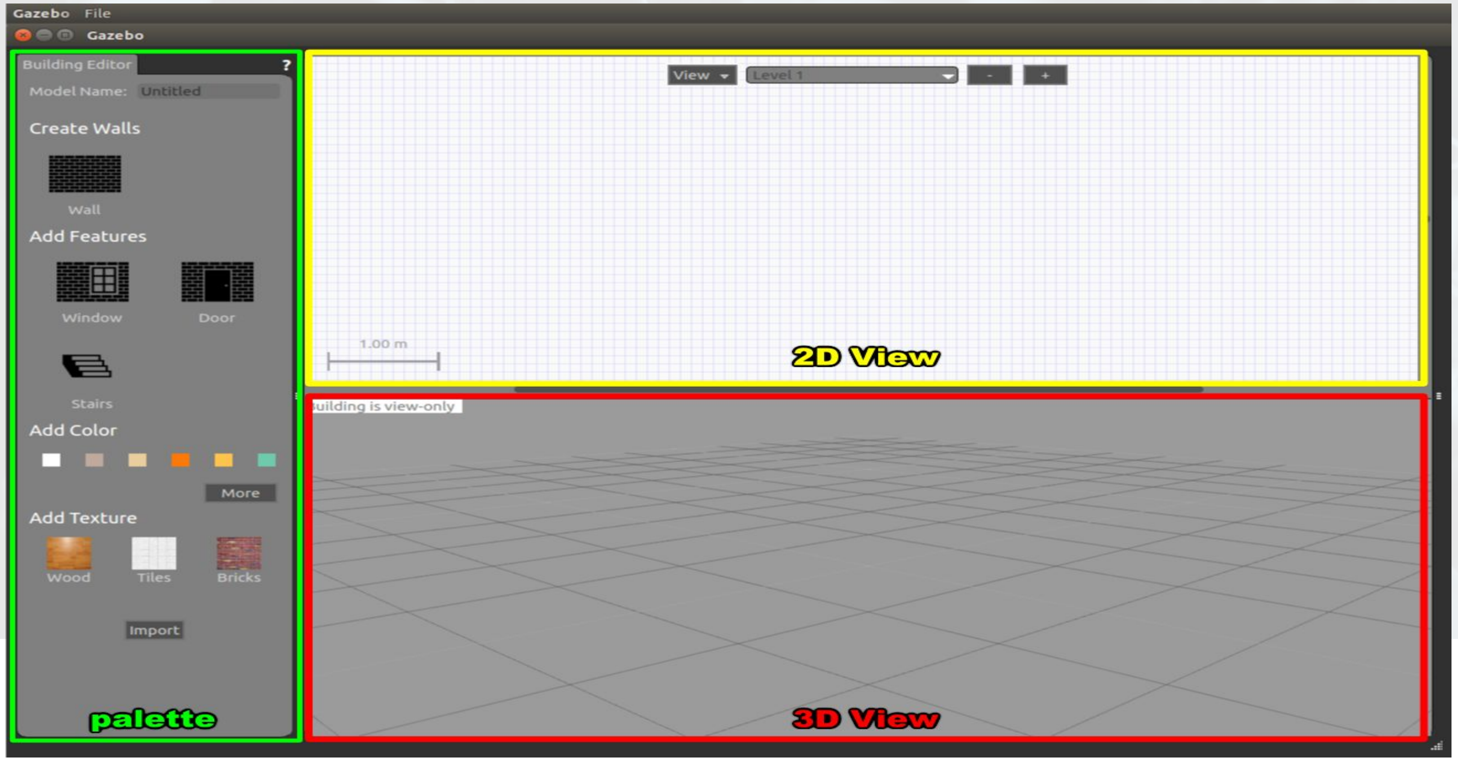
1. **Toolbar** - Contains tools for editting the model
2. **Palette** - Also known as *Left Panel*. Has two tabs for editing the model.
3. **Insert tab** - Tools for adding links and nested models
4. **Model tab** - Allows editing model properties and contents

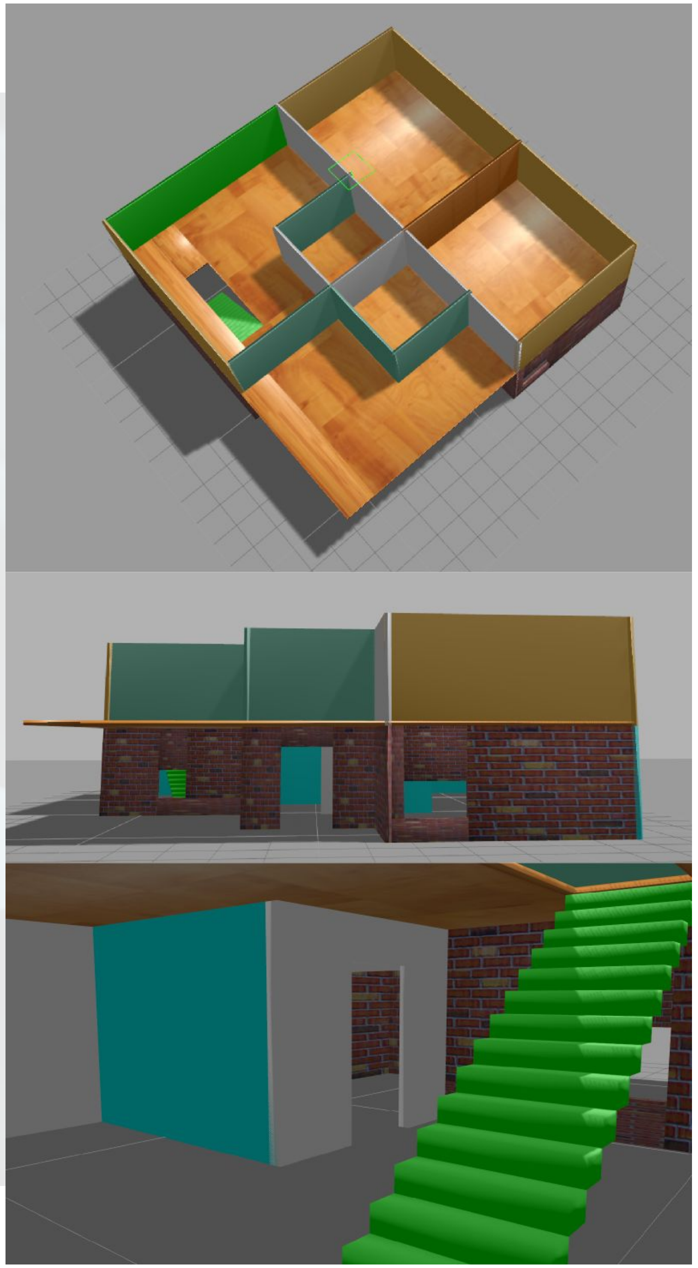
NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

COMPANY CONFIDENTIAL

Building Editor

- The **Palette**, where you can choose features and materials for your building.
- The **2D View**, where you can import a floor plan to trace over (optional) and insert walls, windows, doors and stairs.
- The **3D View**, where you can see a preview of your building. It is also where you can assign colors and textures to different parts of your building.





NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

COMPANY CONFIDENTIAL

ROS INTEGRATION

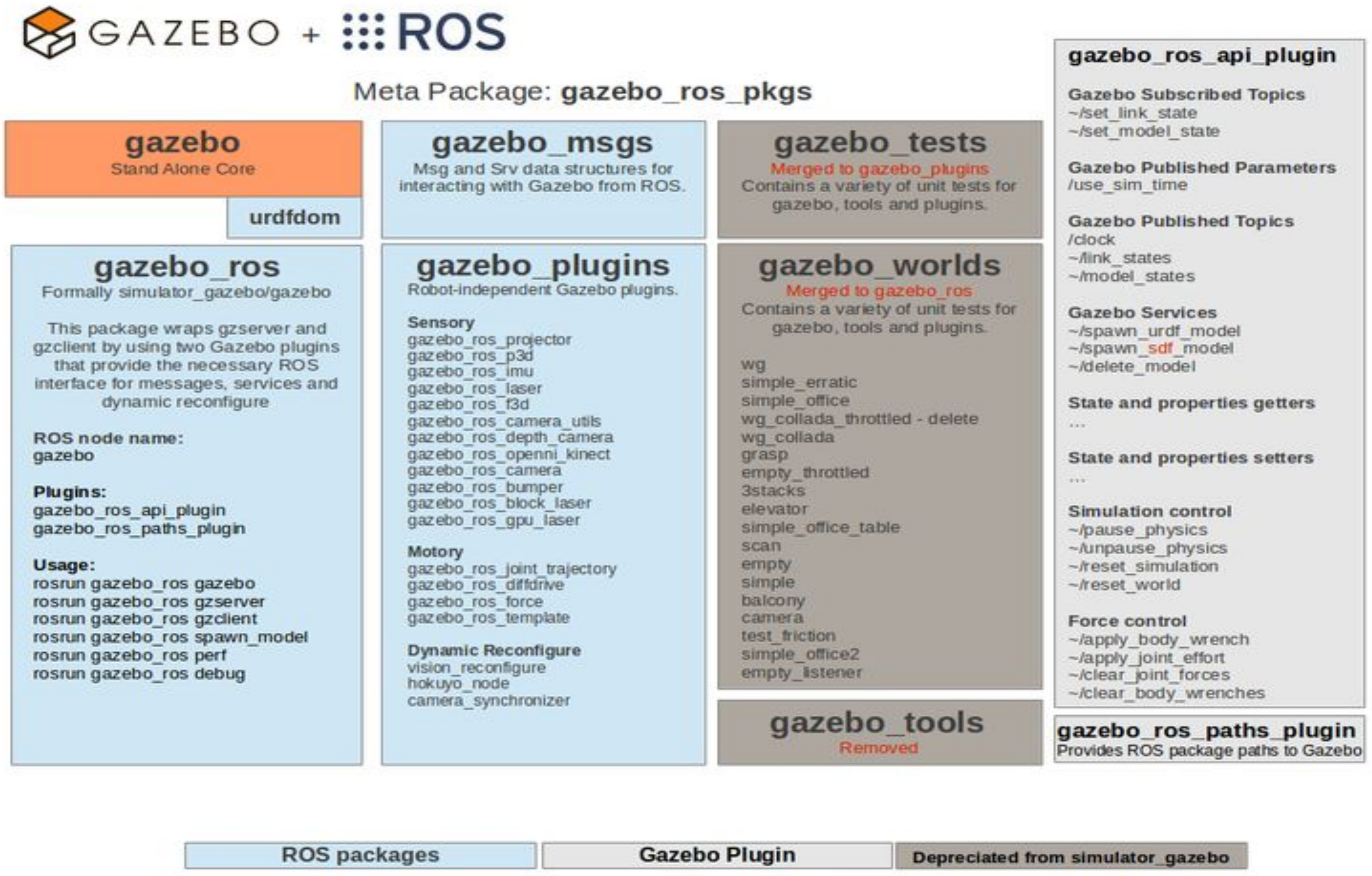
Gazebo is a stand-alone application which can be used independently of ROS or ROS 2. The integration of Gazebo with either ROS version is done through a set of packages called gazebo_ros_pkgs. These packages provide a bridge between Gazebo's C++ API and transport system, and ROS messages and services.

Some features of gazebo_ros_pkgs:

- Supports a stand alone system dependency of Gazebo, that has no ROS bindings on its own
- Builds with catkin
- Treats URDF and SDF as equally as possible
- Reduces code duplication with Gazebo
- Improves out of the box support for controllers using ros_control
- Integrates real time controller efficiency improvements from the DARPA Robotics Challenge
- Cleans up old code from previous versions of ROS and Gazebo

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

COMPANY CONFIDENTIAL



GAZEBO + ROS

Meta Package: **gazebo_ros_pkgs**

gazebo
Stand Alone Core

urdfdom

gazebo_msgs
Msg and Srv data structures for interacting with Gazebo from ROS.

gazebo_tests
Merged to gazebo_plugins
Contains a variety of unit tests for gazebo, tools and plugins.

gazebo_ros
Formally simulator_gazebo/gazebo

This package wraps gzserver and gzclient by using two Gazebo plugins that provide the necessary ROS interface for messages, services and dynamic reconfigure

ROS node name:
gazebo

Plugins:
gazebo_ros_api_plugin
gazebo_ros_paths_plugin

Usage:
rosrun gazebo_ros gazebo
rosrun gazebo_ros gzserver
rosrun gazebo_ros gzclient
rosrun gazebo_ros spawn_model
rosrun gazebo_ros perf
rosrun gazebo_ros debug

gazebo_plugins
Robot-independent Gazebo plugins.

Sensory
gazebo_ros_projector
gazebo_ros_p3d
gazebo_ros_imu
gazebo_ros_laser
gazebo_ros_f3d
gazebo_ros_camera_utils
gazebo_ros_depth_camera
gazebo_ros_openni_kinect
gazebo_ros_camera
gazebo_ros_bumper
gazebo_ros_block_laser
gazebo_ros_gpu_laser

Motory
gazebo_ros_joint_trajectory
gazebo_ros_diffdrive
gazebo_ros_force
gazebo_ros_template

Dynamic Reconfigure
vision_reconfigure
hokuyo_node
camera_synchronizer

gazebo_worlds
Merged to gazebo_ros
Contains a variety of unit tests for gazebo, tools and plugins.

wg
simple_erratic
simple_office
wg_collada_throttled - delete
wg_collada
grasp
empty_throttled
3stacks
elevator
simple_office_table
scan
empty
simple
balcony
camera
test_friction
simple_office2
empty_listener

gazebo_tools
Removed

gazebo_ros_api_plugin

Gazebo Subscribed Topics
~/set_link_state
~/set_model_state

Gazebo Published Parameters
/use_sim_time

Gazebo Published Topics
/clock
~/link_states
~/model_states

Gazebo Services
~/spawn_urdf_model
~/spawn_sdf_model
~/delete_model

State and properties getters
...

State and properties setters
...

Simulation control
~/pause_physics
~/unpause_physics
~/reset_simulation
~/reset_world

Force control
~/apply_body_wrench
~/apply_joint_effort
~/clear_joint_forces
~/clear_body_wrenches

gazebo_ros_paths_plugin
Provides ROS package paths to Gazebo

ROS packages | Gazebo Plugin | Depreciated from simulator_gazebo

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.

COMPANY CONFIDENTIAL

SECURE CONNECTIONS
FOR A SMARTER WORLD

Thank you

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V. ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.