Gazebo Tutorials

October 19, 2012
Willow Garage

Welcome

- Introductions
- Setup and usage
- Core concepts
- Tutorials
    - Building a mobile robot
    - Controlling a mobile robot
    - Building a world
    - ROS integration
    - DRC Simulator

Setup and Usage

Setup and Help

Install

- Does everyone have Gazebo installed?
- Can everyone run Gazebo?

```
$ gazebo
```

Help

- answers.gazebosim.org

  Ask questions and find answers to Gazebo problems
- gazebosim.org/wiki

  Tutorials, and user maintained documentation
- gazebosim.org/api

  Doxygen generated code documentation
- gazebosim.org/user_guide

  Written guide to using Gazebo

The Graphical Interface

Design Principles

- Balance between world navigation and model centric navigation
- Make the most common actions easily accessible

System Perspective

- Gazebo consists of two parts

Server: physics engine, sensor generation

Client(s): Graphical interface, command line tools, your custom application

New in Version 1.2

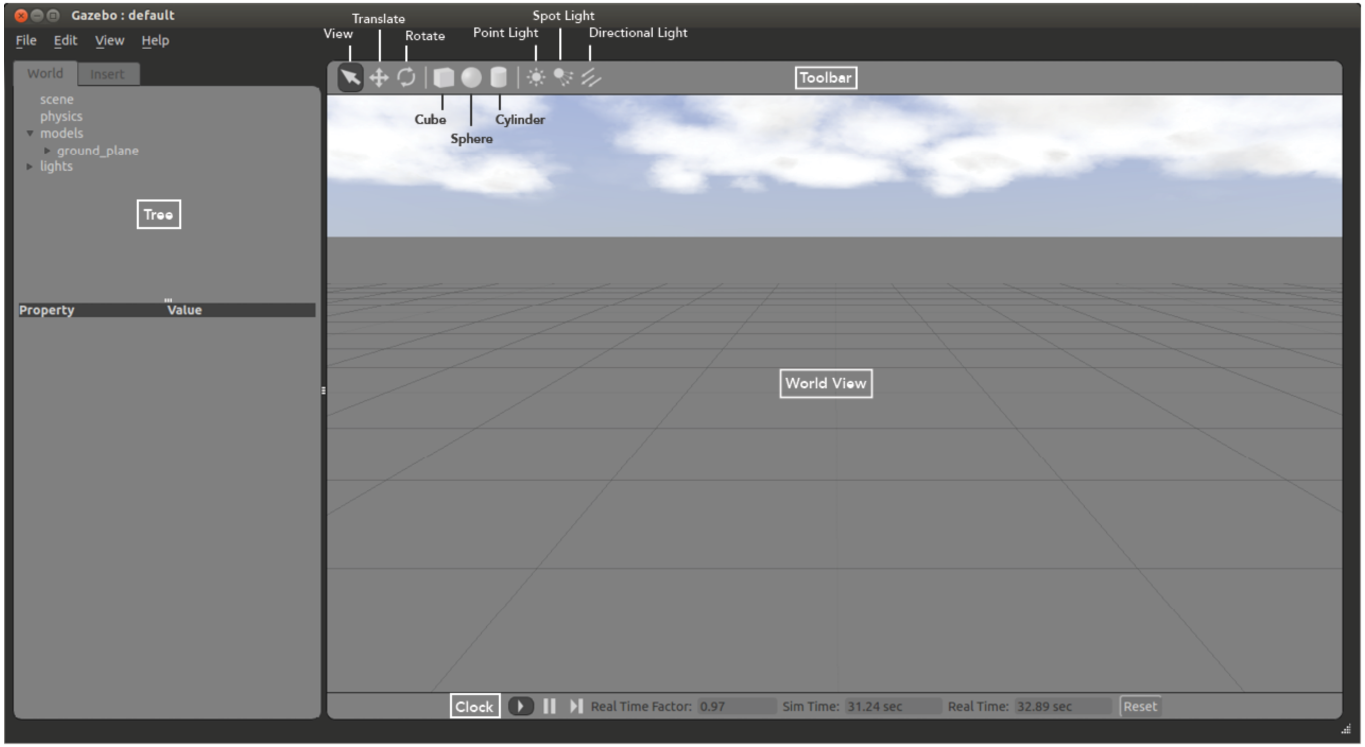
- Simplified mouse controls
- Drag-and-drop models
- Consolidated simulation information
- New style

Using the Graphical Interface

GUI Documentation

http://gazebosim.org/user_guide/started_gui.html

Everyone: start Gazebo



Core Concepts

Elements within Simulation

World

- Collection of models, lights, plugins and global properties

Models

- Collection of links, joints, sensors, and plugins

Links

- Collection of collision and visual objects

Collision Objects

- Geometry that defines a colliding surface

Visual Objects

- Geometry that defines visual representation

Joints

- Constraints between links

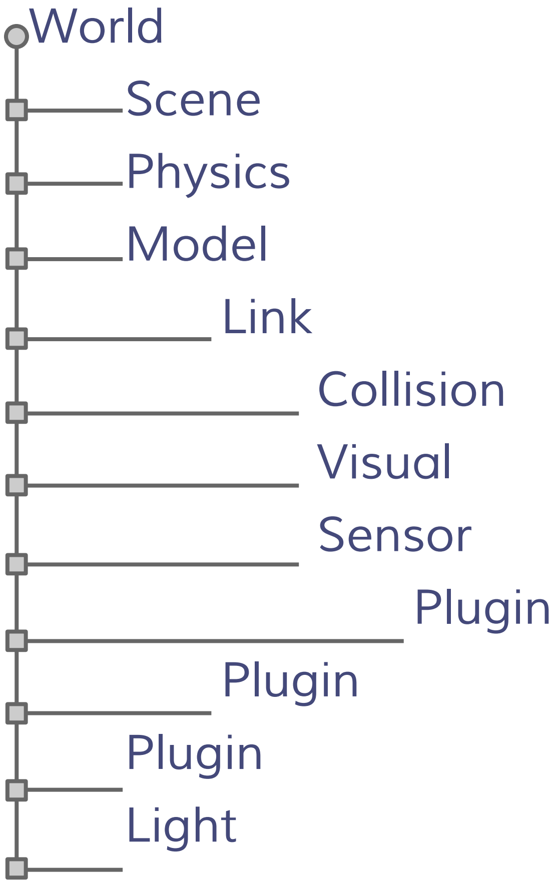
Sensors

- Collect, process, and output data

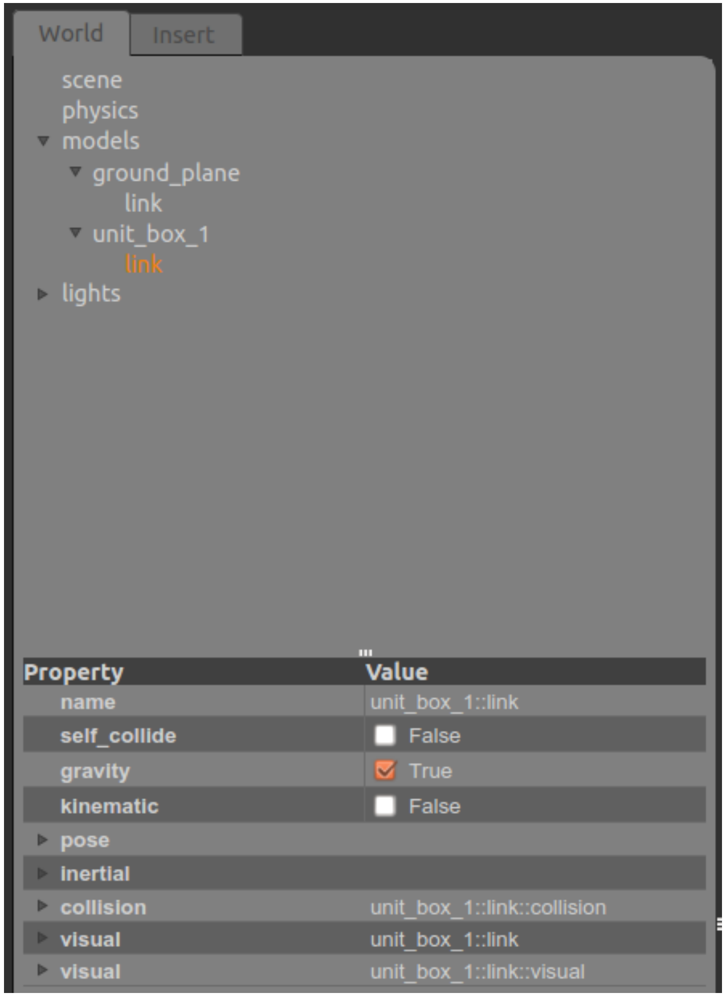
Plugins

- Code attached to a World, Model, Sensor, or the simulator itself

Element Hierarchy



World
- Scene
- Physics
- Model
  - Link
    - Collision
    - Visual
    - Sensor
      - Plugin
  - Plugin
- Plugin
- Light



- scene
- physics
- ▼ models
  - ▼ ground_plane
    - link
  - ▼ unit_box_1
    - link
- ▶ lights

Property	Value
name	unit_box_1::link
self_collide	□ False
gravity	☑ True
kinematic	□ False
▶ pose	
▶ inertial	
▶ collision	unit_box_1::link::collision
▶ visual	unit_box_1::link
▶ visual	unit_box_1::link::visual

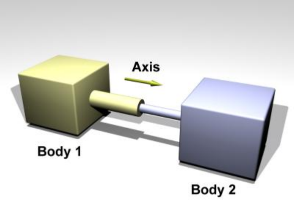
Element Types

Collision and Visual Geometries

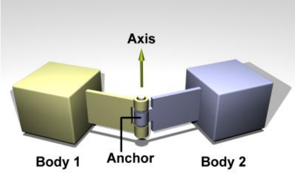
- Simple shapes: sphere, cylinder, box, plane
- Complex shapes: heightmaps, meshes

Joints

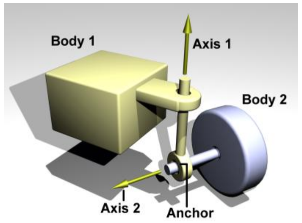
- Prismatic: 1 DOF translational
- Revolute: 1 DOF rotational
- Revolute2: Two revolute joints in series
- Ball: 3 DOF rotational
- Universal: 2 DOF rotational
- Screw: 1 DOF translational, 1 DOF rotational



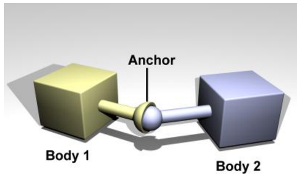
Prismatic



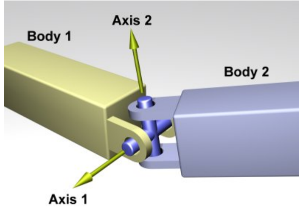
Revolute



Revolute 2



Ball



Universal

Element Types

Sensors

- Ray: produces range data
- Camera (2D and 3D): produces image and/or depth data
- Contact: produces collision data
- RFID: detects RFID tags

Lights

- Point: omni-directional light source, a light bulb
- Spot: directional cone light, a spot light
- Directional: parallel directional light, sun

Exercise 1: Building a mobile robot

Exercise 1
Overview

Topics Covered

- Construction of a two-wheeled mobile base
- Attaching meshes to visual elements
- Attaching sensors to links
- Constructing a simple gripper
- Attaching a gripper to a mobile base

Wiki Tutorials

http://gazebosim.org/wiki/Tutorials

Section: Building a Robot

Exercise 2: Controlling a mobile robot

Exercise 2
Simulation Controls Overview

Animation vs. Dynamic control

- Animation

  Fast.

  Disregard physics, constraints*.

  No collision responses.

- Dynamic control

  Velocity control - leveraging integrator only

  Force control - leveraging physics engine (f = ma)

  Can be computationally intensive

- Controllers with sensor feedback.
- Gazebo's built-in PID class.

Exercise 2
Simulation Controls Overview

Topics Covered

- Animating pose of rigid body links with the animation engine.
- Controlling pose of rigid body links by setting velocities.
- Controlling joints by applying forces.
- Controlling a robot with its simulated onboard sensor.
- Controlling a joint with Gazebo's builtin PID class.

Wiki Tutorials

http://gazebosim.org/wiki/Tutorials

Section: Controlling a Robot

Exercise 3: Building a world

Exercise 3
Overview

Topics Covered

- Constructing a world using the graphical interface
- Modifying world parameters
- Controlling the world via a plugin

Wiki Tutorials

http://gazebosim.org/wiki/Tutorials

Section: Making a World

Exercise 4: ROS Integration

Exercise 4
ROS Integration Overview

Gazebo in ROS or ROS in Gazebo?

- ROS wrapped thirdparty Gazebo installation (http://ros.org/wiki/simulator_gazebo)

  Fuerte ⟵ Gazebo 1.0.x

- Gazebo standalone installation (http://gazebosim.org)

Model Description Formats: COLLADA, URDF, SDF, SRDF, YADF?

- Solidworks to URDF exporter

  http://ros.org/wiki/sw_urdf_exporter

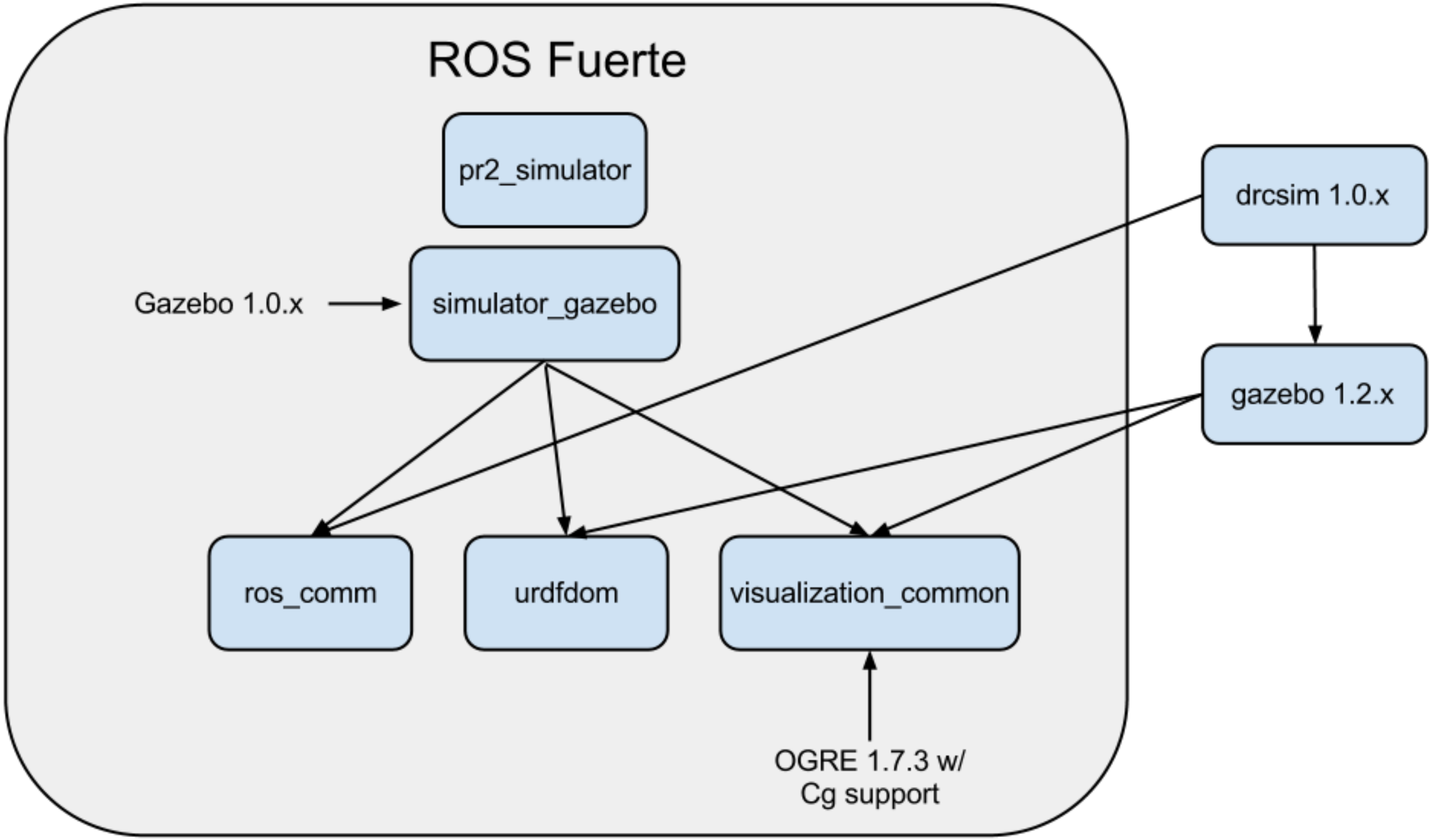
- URDF Dependencies

  http://ros.org/wiki/urdf

  URDF support built at compile time in Gazebo 1.2.x*

  ```
  sudo apt-get install ros-fuerte-urdfdom
  ```

Exercise 4
ROS Integration Overview



Exercise 4
ROS Integration Overview

Gazebo Plugins with ROS dependencies

- For simulating ROS drivers for real robots

    http://ros.org/wiki/wge100_camera_firmware

    http://ros.org/wiki/microstrain_3dmgx2_imu

    http://ros.org/wiki/prosilica

    ...

- Using high level ROS applications with Gazebo

    http://ros.org/wiki/navigation

    http://ros.org/wiki/pr2_interactive_manipulation

    http://moveit.ros.org

    ...

Exercise 4
ROS Integration Overview

Topics Covered

- Managing ROS dependencies
- Building a Gazebo plugin with ROS

Wiki Tutorials

http://gazebosim.org/wiki/Tutorials

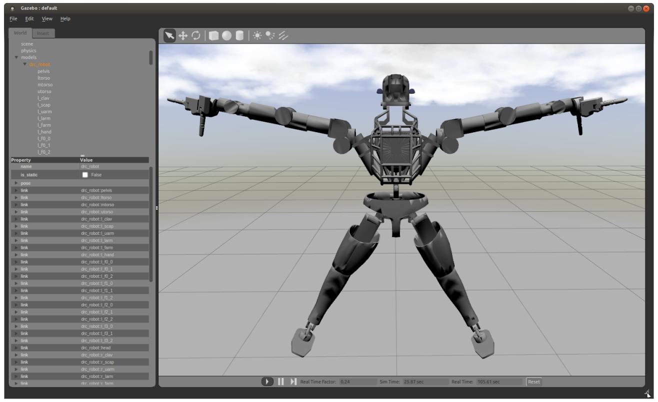
Section: ROS Integration

Exercise 5: DRC Simulator

Exercise 5
DRC Robot Overview

DRC Robot Dynamics Model

- Initial URDF generated form simplified CAD model subject to change.



scene
physics
models
drc_robot
pelvis
ltorso
mtorso
utorso
l_clav
l_scap
l_uarm
l_larm
l_farm
l_hand
l_f0_0
l_f0_1
l_f0_2

Property	Value
name	drc_robot
is_static	☐ False
pose	
link	drc_robot::pelvis
link	drc_robot::ltorso
link	drc_robot::mtorso
link	drc_robot::utorso
link	drc_robot::l_clav
link	drc_robot::l_scap
link	drc_robot::l_uarm
link	drc_robot::l_larm
link	drc_robot::l_farm
link	drc_robot::l_hand
link	drc_robot::l_f0_0
link	drc_robot::l_f0_1
link	drc_robot::l_f0_2
link	drc_robot::l_f1_0
link	drc_robot::l_f1_1
link	drc_robot::l_f1_2
link	drc_robot::l_f2_0
link	drc_robot::l_f2_1
link	drc_robot::l_f2_2
link	drc_robot::l_f3_0
link	drc_robot::l_f3_1
link	drc_robot::l_f3_2
link	drc_robot::head
link	drc_robot::r_clav
link	drc_robot::r_scap
link	drc_robot::r_uarm
link	drc_robot::r_larm
link	drc_robot::r_farm

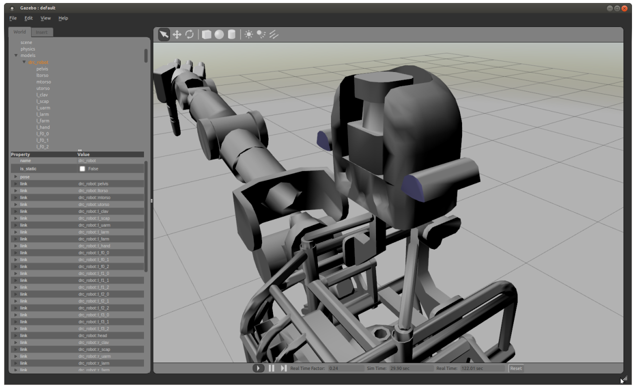
Real Time Factor: 0.24 Sim Time: 25.87 sec Real Time: 105.61 sec

DRC Robot Sensor Suite

- Real sensor suite hardware TBD.
- For now, "Best guess" sensor suite.

    Hokuyo laser

    Stereo camera



scene
physics
models
drc_robot
pelvis
ltorso
mtorso
utorso
l_clav
l_scap
l_uarm
l_larm
l_farm
l_hand
l_f0_0
l_f0_1
l_f0_2

Property	Value
name	drc_robot
is_static	☐ False
pose	
link	drc_robot::pelvis
link	drc_robot::ltorso
link	drc_robot::mtorso
link	drc_robot::utorso
link	drc_robot::l_clav
link	drc_robot::l_scap
link	drc_robot::l_uarm
link	drc_robot::l_larm
link	drc_robot::l_farm
link	drc_robot::l_hand
link	drc_robot::l_f0_0
link	drc_robot::l_f0_1
link	drc_robot::l_f0_2
link	drc_robot::l_f1_0
link	drc_robot::l_f1_1
link	drc_robot::l_f1_2
link	drc_robot::l_f2_0
link	drc_robot::l_f2_1
link	drc_robot::l_f2_2
link	drc_robot::l_f3_0
link	drc_robot::l_f3_1
link	drc_robot::l_f3_2
link	drc_robot::head
link	drc_robot::r_clav
link	drc_robot::r_scap
link	drc_robot::r_uarm
link	drc_robot::r_larm
link	drc_robot::r_farm

Real Time Factor: 0.24 Sim Time: 29.90 sec Real Time: 122.01 sec

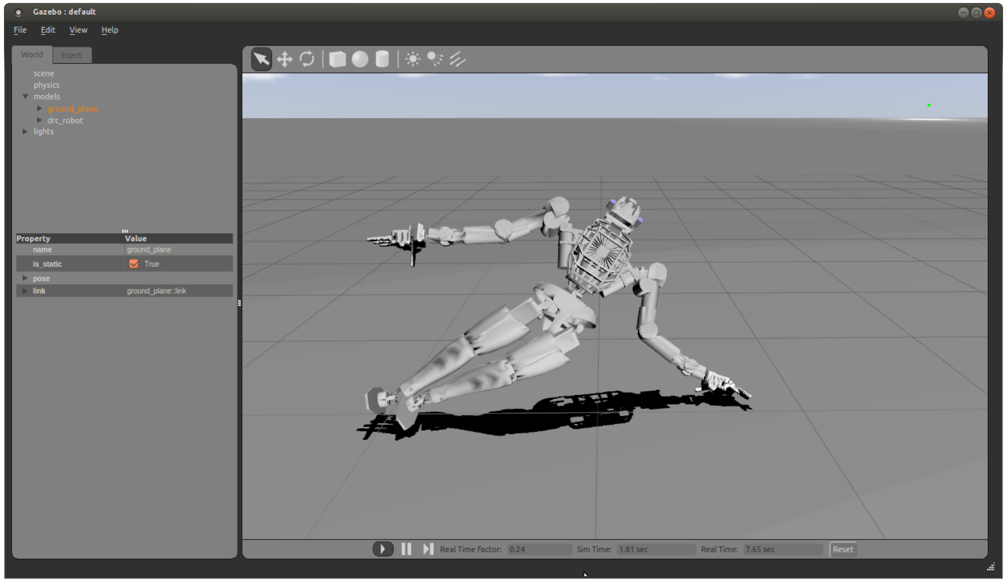
Exercise 5
DRC Robot Overview

DRC Robot Dynamics Controls API

- Initial simulation tutorials "place holder" controllers derived from PR2 controllers

  http://ros.org/wiki/pr2_controllers, http://ros.org/wiki/pr2_mechanism

- Walking controllers interface TBD.



Property	Value
name	ground_plane
is_static	☑ True
pose	
link	ground_plane::link

Real Time Factor: 0.24 Sim Time: 1.81 sec Real Time: 7.65 sec

Exercise 5
DRC Robot Overview

Topics Covered

- Visualize and log sensor data with rviz and rxbag.
- DRC Robot basic joint control using PR2 mechanism controllers.
- Teleporting the DRC Robot.
- Customizing the DRC Robot world contents.
- Animating the DRC Robot with ROS JointTrajectory messages.

  (http://gazebosim.org/wiki/trajectory_msgs)

Wiki Tutorials

http://gazebosim.org/wiki/Tutorials

Section: DRC Tutorials

Exercise 6
Beer and Questions