Back (/tutorials?cat=write_plugin)

Model plugins

Prerequisites

Overview / HelloWorld (http://gazebosim.org/tutorials?tut=plugins_hello_world) Plugin Tutorial

Note: If you're continuing from the previous tutorial, make sure you put in the proper `#include` lines for this tutorial that are listed below.

Code

Source: gazebo/examples/plugins/model_push (https://bitbucket.org/osrf/gazebo/src/gazebo_2.2/examples/plugins/model_push)

Plugins allow complete access to the physical properties of models and their underlying elements (links, joints, collision objects). The following plugin will apply a linear velocity to its parent model.

```
$ cd ~/gazebo_plugin_tutorial
$ gedit model_push.cc
```

Plugin Code:

```
#include <boost/bind.hpp>
#include <gazebo/gazebo.hh>
#include <gazebo/physics/physics.hh>
#include <gazebo/common/common.hh>
#include <stdio.h>

namespace gazebo
{
  class ModelPush : public ModelPlugin
  {
    public: void Load(physics::ModelPtr _parent, sdf::ElementPtr /*_sdf*/)
    {
      // Store the pointer to the model
      this->model = _parent;

      // Listen to the update event. This event is broadcast every
      // simulation iteration.
      this->updateConnection = event::Events::ConnectWorldUpdateBegin(
          boost::bind(&ModelPush::OnUpdate, this, _1));
    }

    // Called by the world update start event
    public: void OnUpdate(const common::UpdateInfo & /*_info*/)
    {
      // Apply a small linear velocity to the model.
      this->model->SetLinearVel(math::Vector3(.03, 0, 0));
    }

    // Pointer to the model
    private: physics::ModelPtr model;

    // Pointer to the update event connection
    private: event::ConnectionPtr updateConnection;
  };

  // Register this plugin with the simulator
  GZ_REGISTER_MODEL_PLUGIN(ModelPush)
}
```

Compiling the Plugin

Assuming the reader has gone through the Hello WorldPlugin tutorial (http://gazebosim.org/tutorials?tut=plugins_hello_world) all that needs to be done is to add the following lines to `~/gazebo_plugin_tutorial/ CMakeLists .txt`

```
add_library(model_push SHARED model_push.cc)
target_link_libraries(model_push ${GAZEBO_LIBRARIES} ${Boost_LIBRARIES})
```

Compiling this code will result in a shared library, `~/gazebo_plugin_tutorial/ build/libmodel_push .so` , that can be inserted in a Gazebo simulation.

```
$ cd ~/gazebo_plugin_tutorial/build
$ cmake ../
$ make
```

Running the Plugin

This plugin is used in the world file `examples /plugins /model_push /model_push .world` .

```
$ cd ~/gazebo_plugin_tutorial
$ gedit model_push.world
```

```
<?xml version="1.0"?>
<sdf version="1.4">
  <world name="default">

    <!-- Ground Plane -->
    <include>
      <uri>model://ground_plane</uri>
    </include>

    <include>
      <uri>model://sun</uri>
    </include>

    <model name="box">
      <pose>0 0 0.5 0 0 0</pose>
      <link name="link">
        <collision name="collision">
          <geometry>
            <box>
              <size>1 1 1</size>
            </box>
          </geometry>
        </collision>

        <visual name="visual">
          <geometry>
            <box>
              <size>1 1 1</size>
            </box>
          </geometry>
        </visual>
      </link>

      <plugin name="model_push" filename="libmodel_push.so"/>
    </model>
  </world>
</sdf>
```

The hook to attach a plugin to a model is specified at the end of the model element block using:

```
Do not copy. The following is for reference purposes.
<plugin name="model_push" filename="libmodel_push.so"/>
```

Add your library path to the `GAZEBO_PLUGIN_PATH` :

```
$ export GAZEBO_PLUGIN_PATH=$HOME/gazebo_plugin_tutorial/build:$GAZEBO_PLUGIN_PATH
```

To start simulation, run

```
$ cd ~/gazebo_plugin_tutorial/
$ gzserver -u model_push.world
```

The `-u` option starts the server in a paused state.

In a separate terminal, start the gui

```
$ gzclient
```

Click on the play button in the gui to unpause the simulation, and you should see the box move.