

# Modelo car-like + Lidar en CoppeliaSim



Bullet 2.78



Model browser

- components
  - grippers
  - locomotion and p...
  - modifiers
  - sensors
- equipment
- examples
  - block factory
  - furniture
  - household
  - infrastructure
  - nature
  - office items
  - other
  - people
  - robots
    - mobile
    - non-mobile
  - tools
  - vehicles

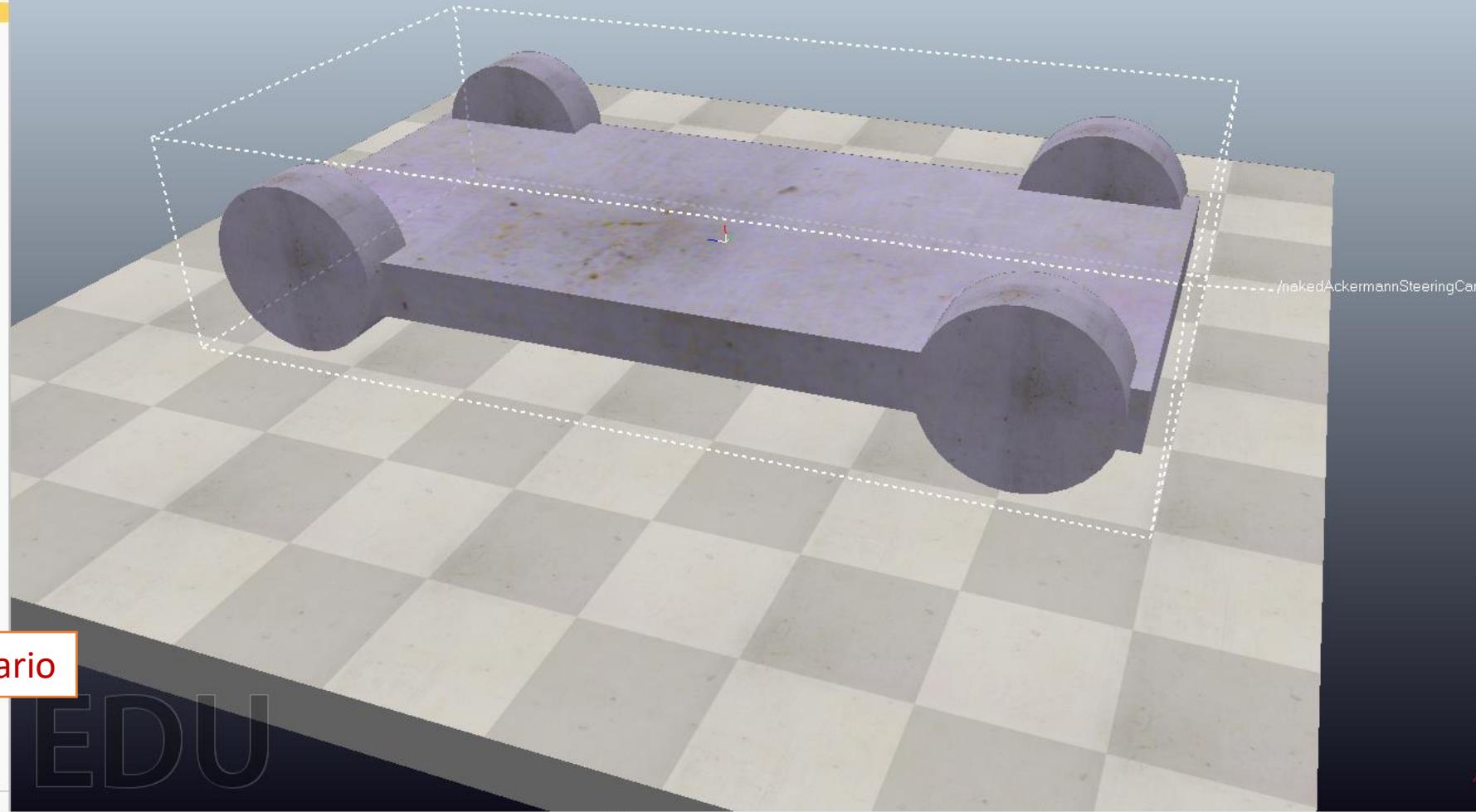
scene\_01

Scene hierarchy

- scene\_01 (scene 1)
  - DefaultCamera
  - XYZCameraProxy
  - DefaultLights
  - Floor

- nakedAckermannSteeringCar

Selected objects:  
1 /nakedAckermannSteeringCar (deprecated name:nakedAckermannSteeringCar)  
Shape (cuboid)  
x: +0.32504 y: +0.275 z: +0.4377  
a: +90.386 b: +0.007 g: +90.003



Mover al escenario

During simulation you can control the car with the up/down and left/right keys (make sure the focus is on the main window)

Input Lua code here, or type "help()" (use TAB for auto-completion)

2 Sandbox script



Bullet 2.78



Model browser

- components
  - grippers
  - locomotion and p...
  - modifiers
  - sensors
- equipment
- examples
  - block factory
  - furniture
  - household
  - infrastructure
  - nature
  - office items
  - other
  - people
  - robots
    - mobile
    - non-mobile
  - tools
  - vehicles

scene\_01

Scene hierarchy

- scene\_01 (scene 1)
  - DefaultCamera
  - XYZCameraProxy
  - DefaultLights
  - Floor
  - nakedAckermannSteeringCar

Selected objects:  
Last selected object alias:  
Last selected object type:  
Last selected object position:  
Last selected object orientation:

1  
/nakedAckermannSteeringCar (deprecated name:nakedAckermannSteeringCar)  
Shape (cuboid)  
x: +0.32504 y: +0.275 z: +0.4377  
a: +90.386 b: +0.007 g: +90.003

Scene Object Properties

Shape      Common

General properties

Selectable       Invisible during selection  
 Select base of model instead       Ignored by depth pass  
 Ignored by model bounding box       Ignored for view-fitting  
 Cannot be deleted during simul.       Cannot be deleted

Extension string

Visibility

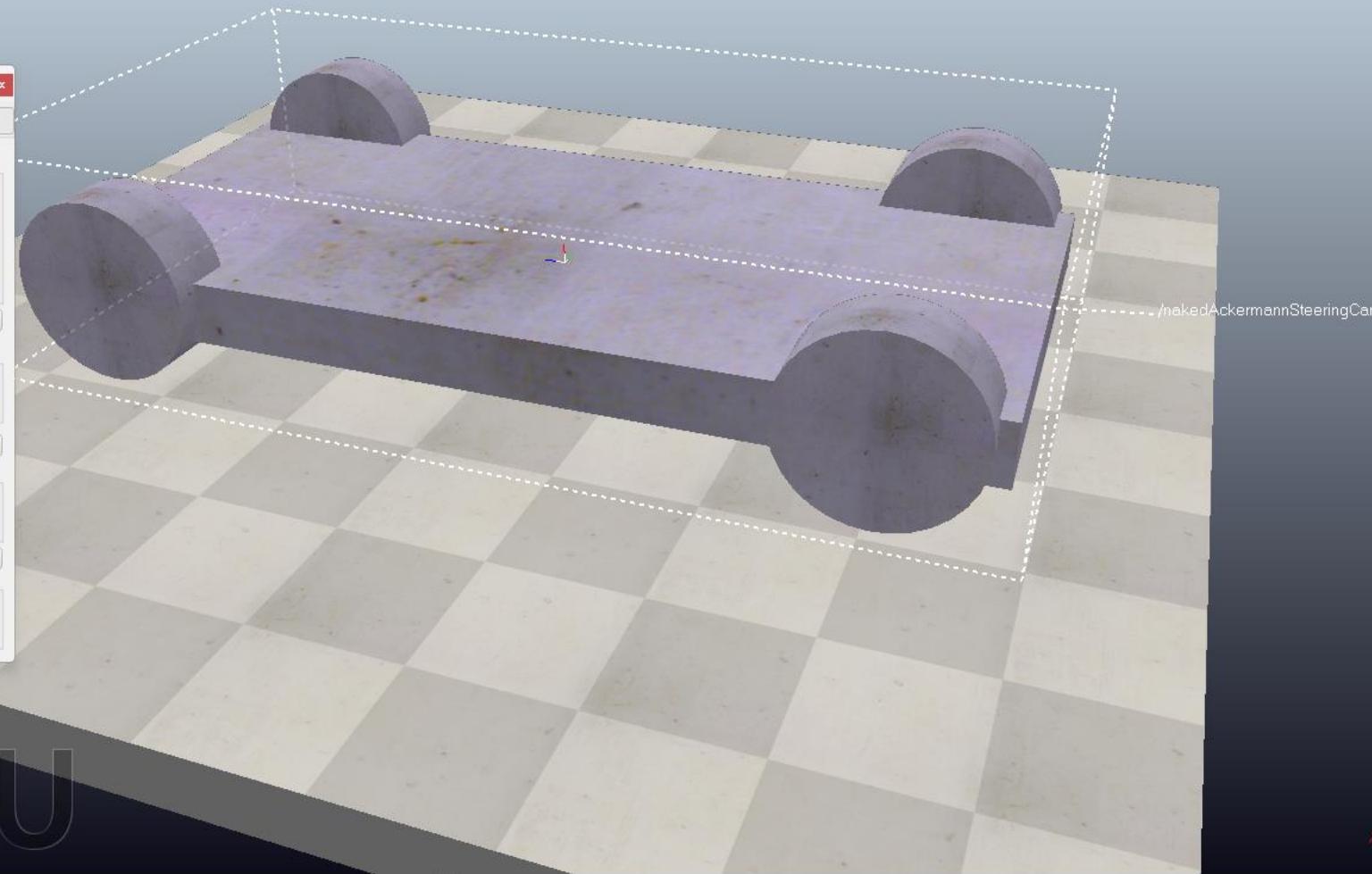
Camera visibility layers

Model definition      Object special properties

Object is model       Collidable       Measurable  
  
 Detectable

Other

Collection self-collision indicator



During simulation you can control the car with the up/down and left/right keys (make sure the focus is on the main window)

Input Lua code here, or type "help()" (use TAB for auto-completion)



Bullet 2.78



Model browser

- components
  - grippers
  - locomotion and p...
  - modifiers
  - sensors
- equipment
- examples
  - block factory
  - furniture
  - household
  - infrastructure
  - nature
  - office items
  - other
  - people
  - robots
    - mobile
    - non-mobile
  - tools
  - vehicles

scene\_01

Scene hierarchy

- scene\_01 (scene 1)
  - DefaultCamera
  - XYZCameraProxy
  - DefaultLights
  - Floor
  - nakedAckermannSteeringCar

Selected objects:  
Last selected object alias:  
Last selected object type:  
Last selected object position:  
Last selected object orientation:

1  
/nakedAckermannSteeringCar (deprecated name:nakedAckermannSteeringCar)  
Shape (cuboid)  
x: +0.32504 y: +0.275 z: +0.4377  
a: +90.386 b: +0.007 g: +90.003

Scene Object Properties

Shape      Common

General properties

Selectable       Invisible during selection  
 Select base of model instead       Ignored by depth pass  
 Ignored by model bounding box       Ignored for view-fitting  
 Cannot be deleted during simul.       Cannot be deleted

Extension string

Object / Model Scaling

In-place scaling       to selection

Visibility

Camera visib      Scaling factor 0.1

OK      Cancel

Apply to selection

Model definition      Object special properties

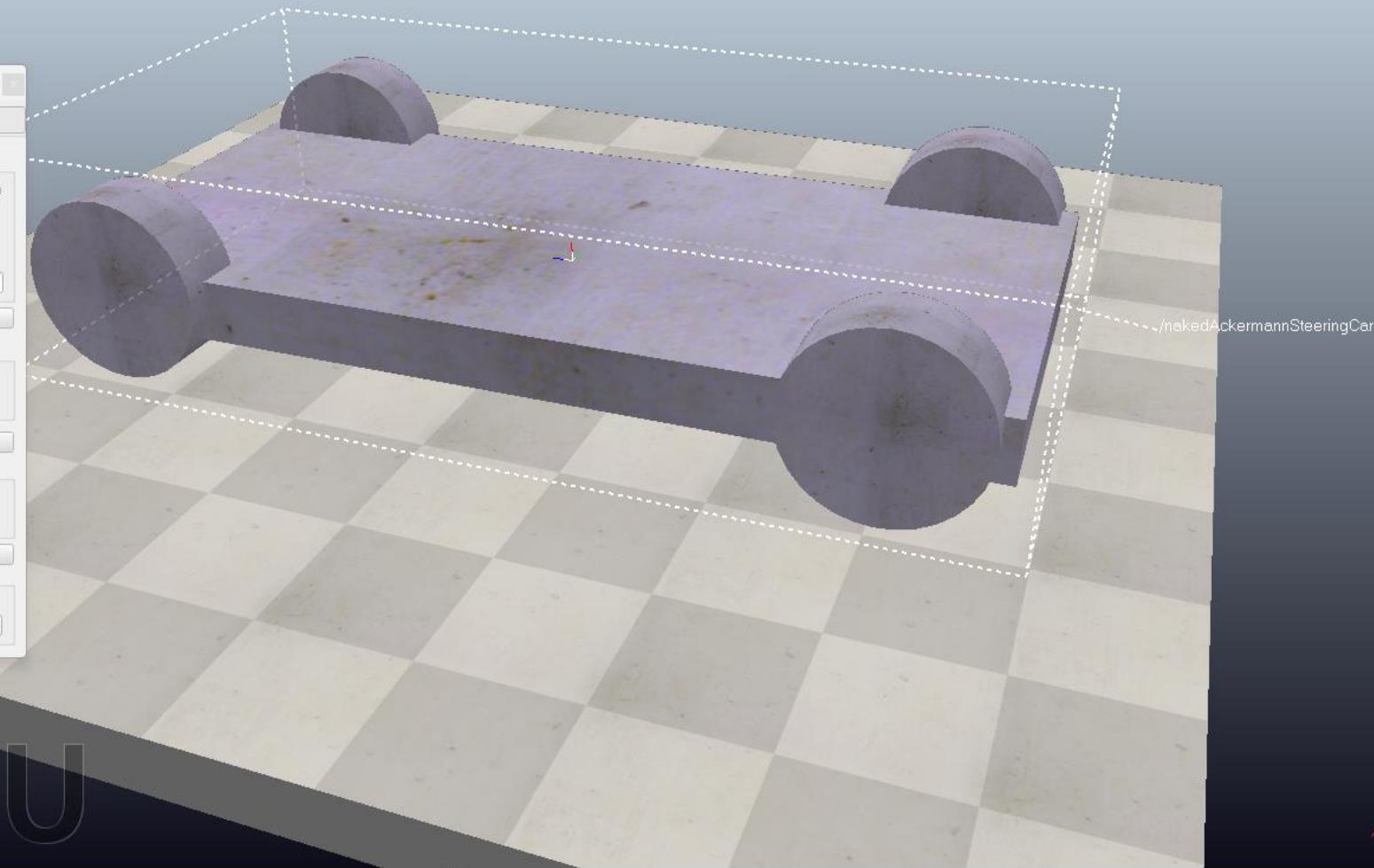
Object is model       Collidable       Measurable  
 Model properties       Detectable

Apply to selection

Other

Collection self-collision indicator 0  
Scaling      Assembling

Apply to selection



block factory



simple Ackermann steering.ttm



[CoppeliaSim:info] Executing undo...  
[CoppeliaSim:info] done.

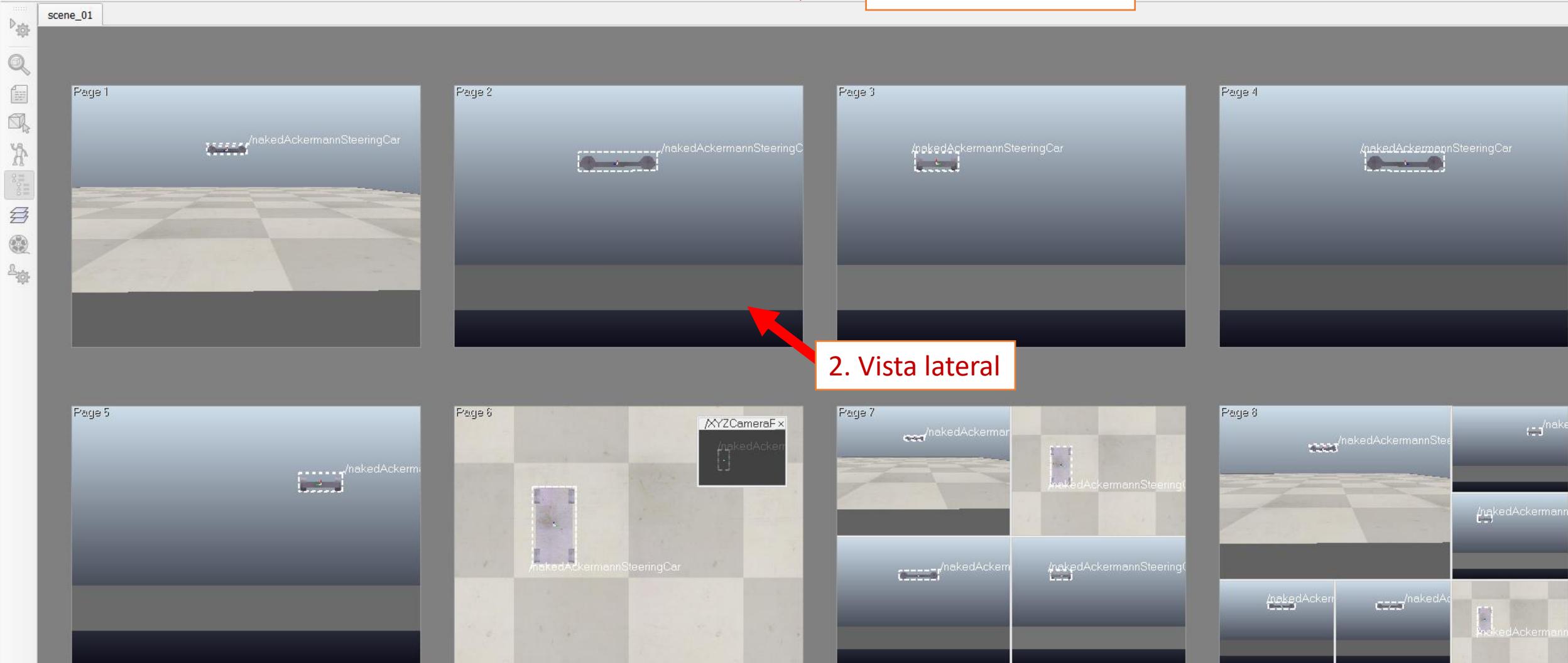
Input Lua code here, or type "help()" (use TAB for auto-completion)



Bullet 2.78



1. Selector de vistas

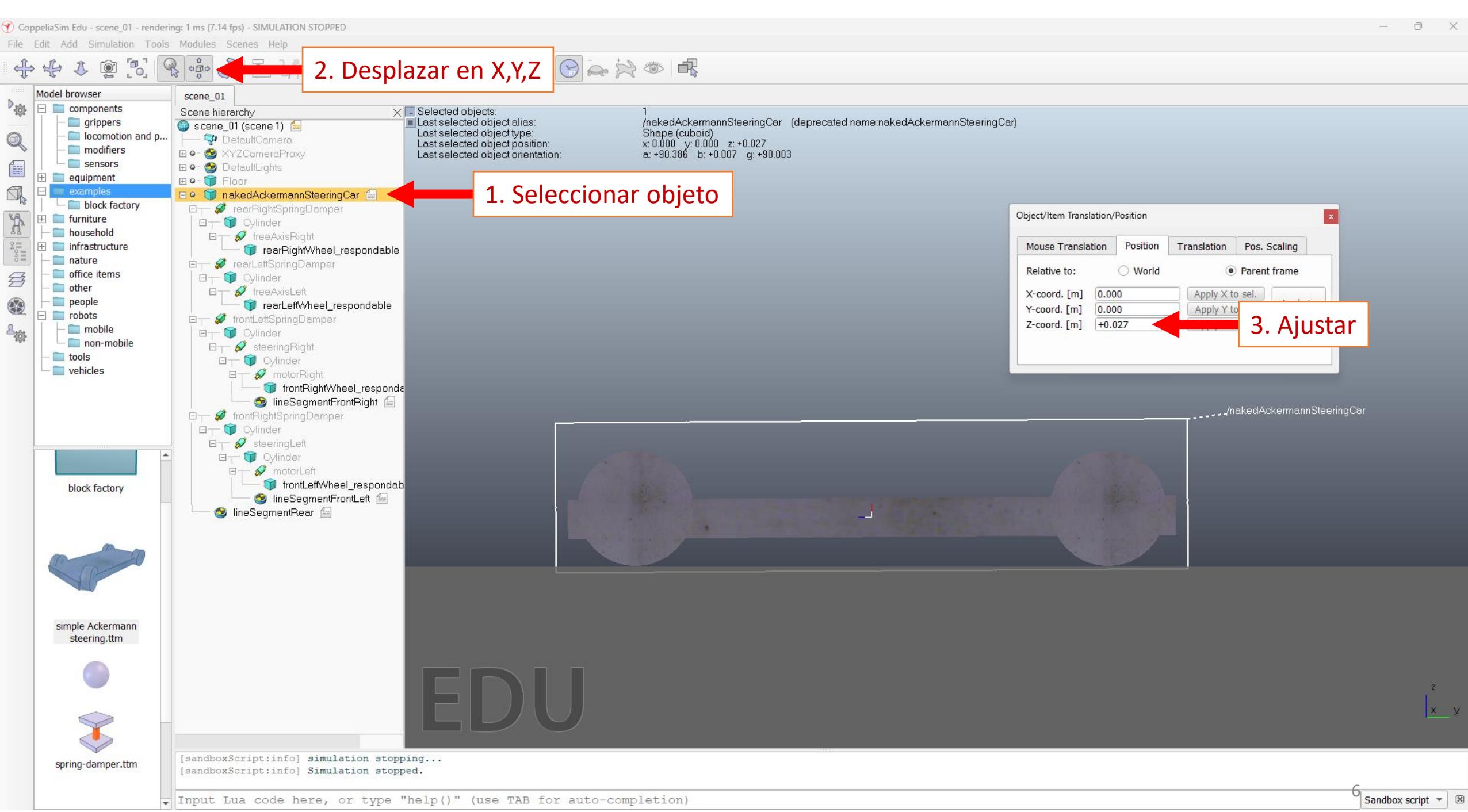


[CoppeliaSim:info] Executing undo...  
[CoppeliaSim:info] done.

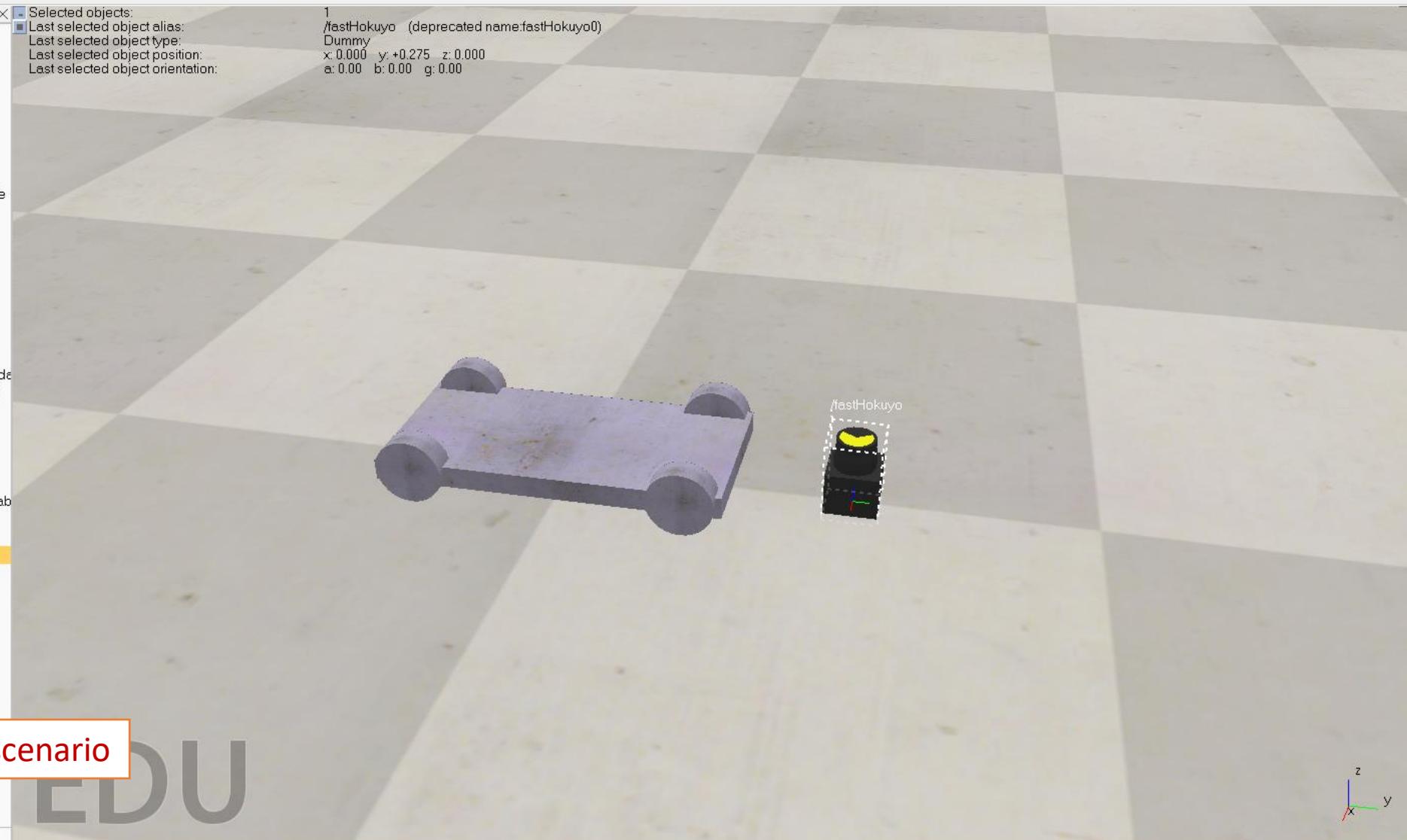
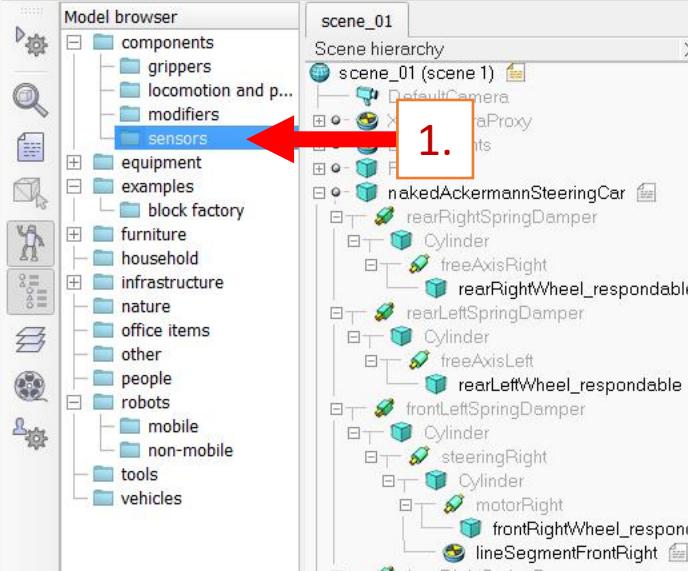
Input Lua code here, or type "help()" (use TAB for auto-completion)

5

Sandbox script



# Sensor Hokuyo en V-Rep



Selected objects:  
Last selected object alias:  
Last selected object type:  
Last selected object position:  
Last selected object orientation:

1 fastHokuyo (deprecated name fastHokuyo0)

Dummy

x: 0.000 y: +0.275 z: 0.000

a: 0.00 b: 0.00 g: 0.00

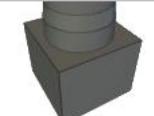
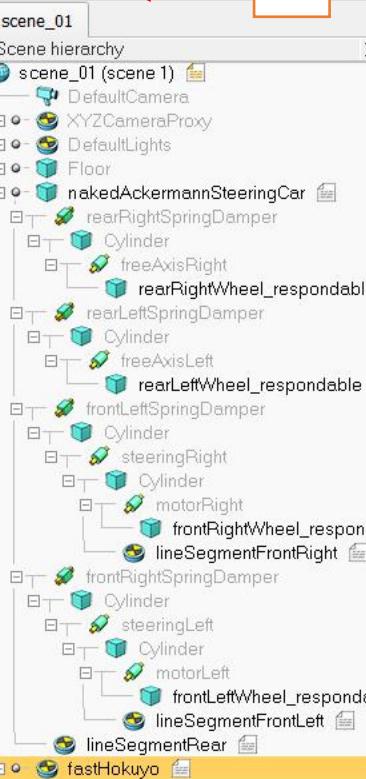
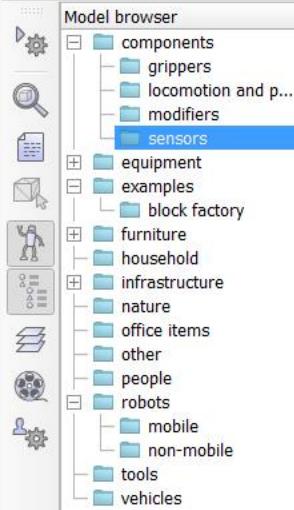
2. Mover al escenario

```
[sandboxScript:info] simulation stopping...
[sandboxScript:info] Simulation stopped.
```

Input Lua code here, or type "help()" (use TAB for auto-completion)

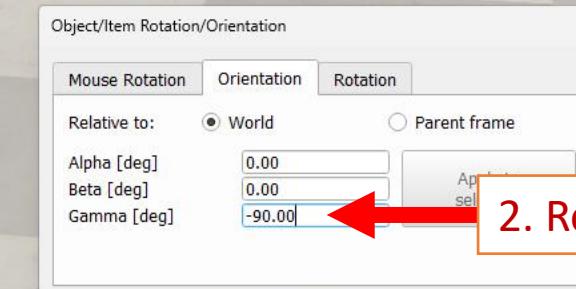
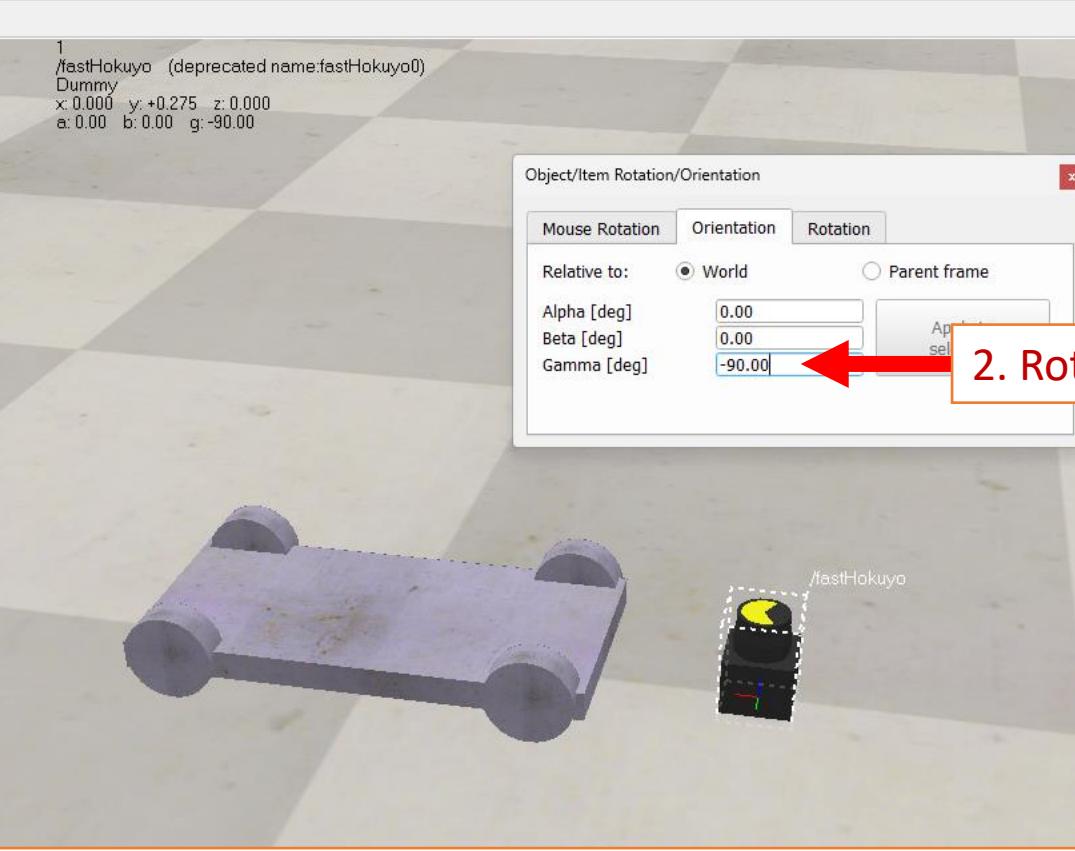


1.

Hokuyo URG 04LX  
UG01\_Fast ROS.ttmHokuyo URG 04LX  
UG01\_Fast.ttm

```
[sandboxScript:info] simulation stopping...
[sandboxScript:info] Simulation stopped.
```

Input Lua code here, or type "help()" (use TAB for auto-completion)

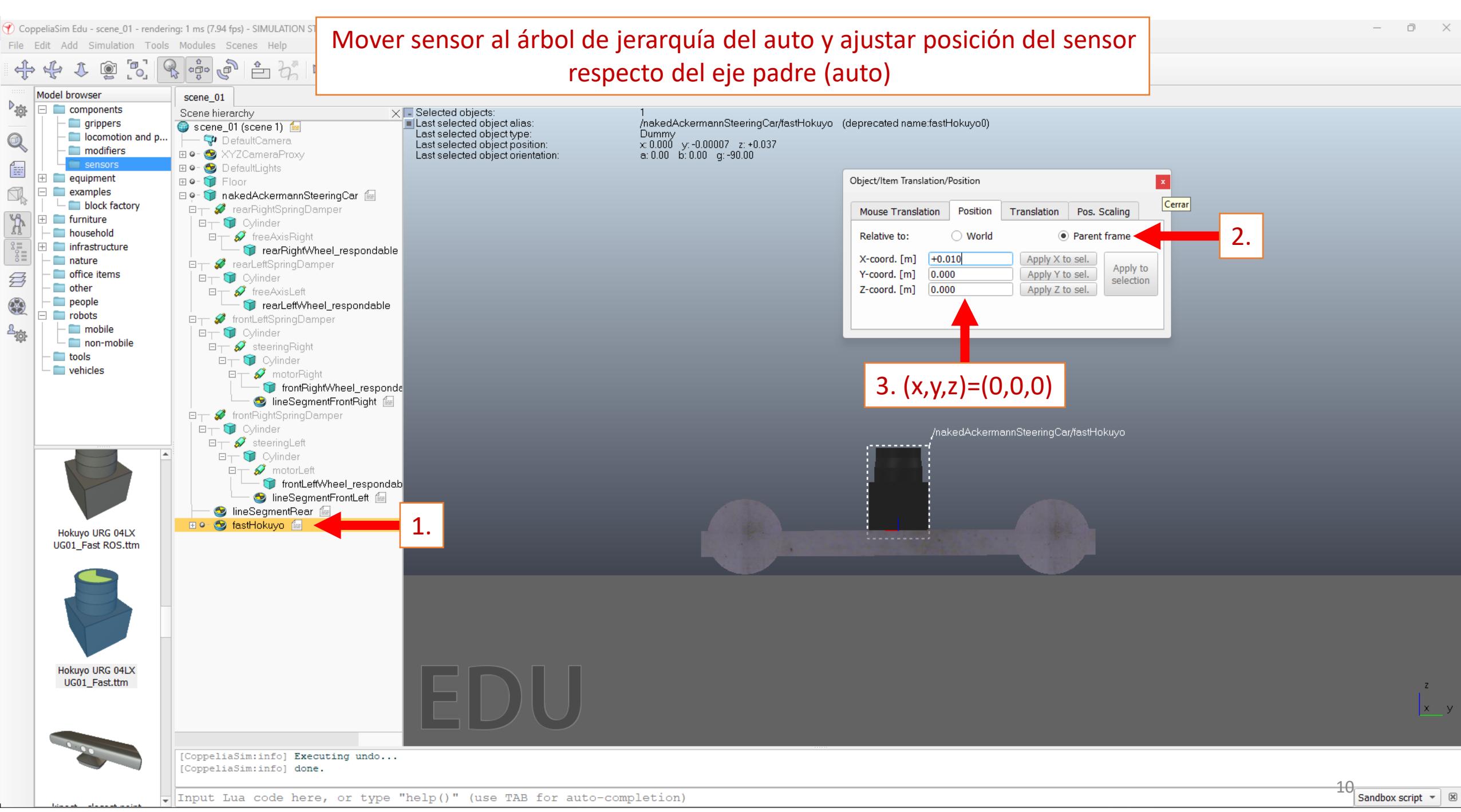


2. Rotar en Z

Nota: Coincidir frame (eje X) del sensor Hokuyo con la dirección “hacia adelante” del auto

EDU

z  
y  
x



# Editando escenario en V-Rep



Bullet 2.78



Model browser

- + components
- + equipment
- + examples
- + furniture
- + household
  - + infrastructure
    - + bathroom
    - + doors
    - floors**
    - + other
    - + urban
  - + walls
- + windows
- + nature
- + office items
- + other
- + people
- + robots
  - + mobile
  - + non-mobile
- + tools
- + vehicles



resizable floor 1-5 meters.ttm



resizable floor 25-100 meters.ttm



resizable floor 5-25 meters.ttm

scene\_01  
Scene hierarchy

scene\_01 (scene 1)  
DefaultCamera  
XYZCameraProxy  
DefaultLights  
nakedAckermannSteeringC

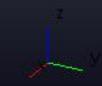
0



Borrar objeto: Floor



EDU



[CoppeliaSim:info] Deleting selection...  
[CoppeliaSim:info] done.

Input Lua code here, or type "help()" (use TAB for auto-completion)



Bullet 2.78

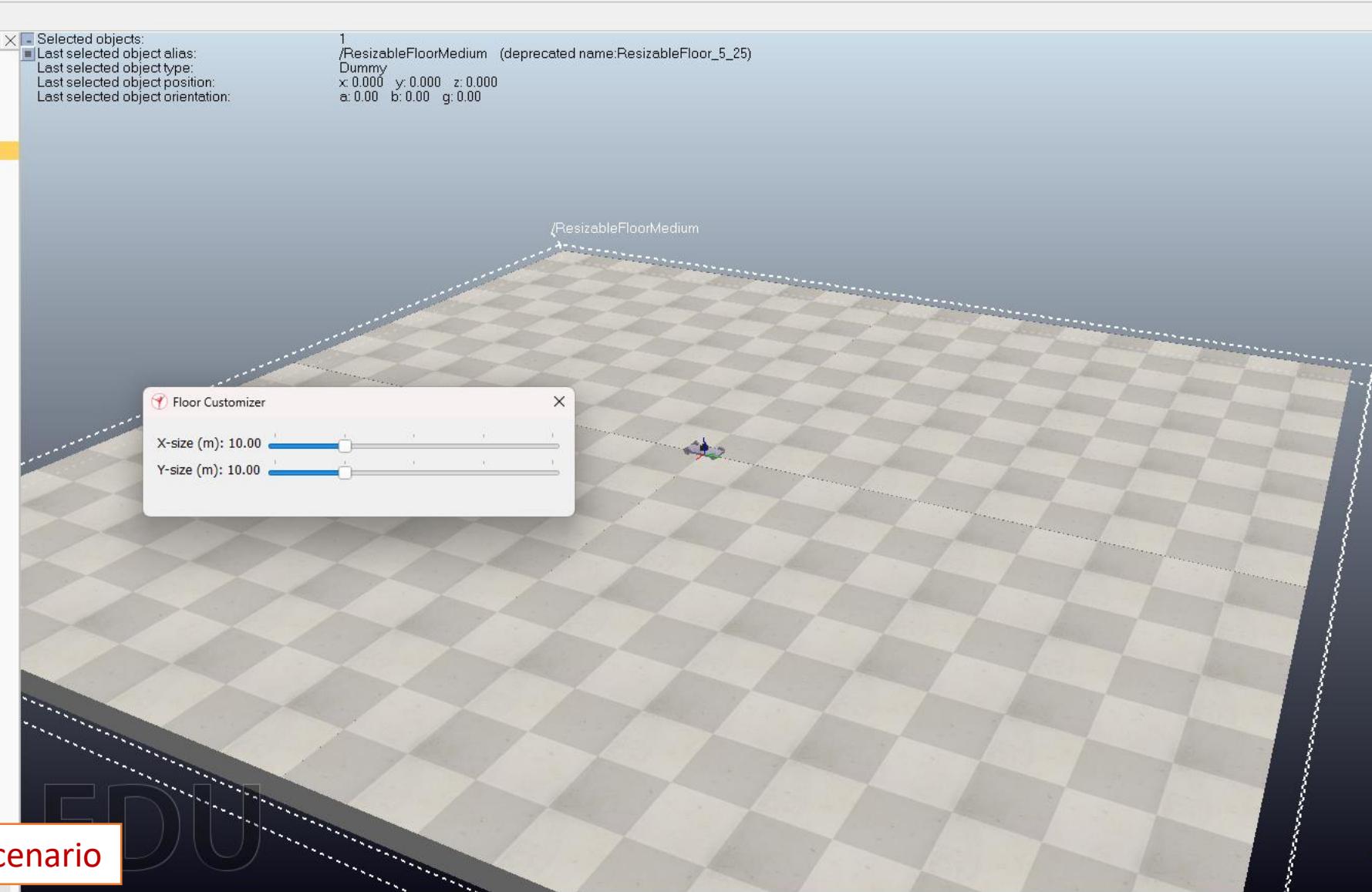
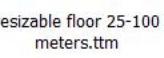


Model browser

- scene\_01
- Scene hierarchy
  - scene\_01 (scene 1)
    - DefaultCamera
    - XYZCameraProxy
    - DefaultLights
    - nakedAckermannSteeringCar
    - ResizableFloorMedium
- Selected objects:
  - 1 /ResizableFloorMedium (deprecated name:ResizableFloor\_5\_25)

1.

- components
- equipment
- examples
- furniture
- household
  - infrastructure
    - bathroom
    - doors
    - floors
    - other
    - urban
    - walls
  - windows
- nature
- office items
- other
- people
- robots
  - mobile
  - non-mobile
- tools
- vehicles



2. Mover al escenario

[CoppeliaSim:info] File was previously written with CoppeliaSim version 4.05.01 (rev 0)  
[CoppeliaSim:info] Model loaded.

Input Lua code here, or type "help()" (use TAB for auto-completion)

Sandbox script



Bullet 2.78



Model browser

- scene\_01
- Scene hierarchy
  - scene\_01 (scene 1)
    - DefaultCamera
    - XYZCameraProxy
    - DefaultLights
    - nakedAckermannSteeringCar
    - ResizableFloorMedium
    - ExternalWall

Selected objects:

- Last selected object alias: /ExternalWall (deprecated name:ExternalWall)
- Dummy
- Last selected object position: x: +0.050 y: -0.800 z: 0.000
- Last selected object orientation: a: 0.00 b: 0.00 g: 0.00

1.  

2. Mover al escenario  

240cm high walls

80cm high walls

external wall.ttm

[CoppeliaSim:info] File was previously written with CoppeliaSim version 4.05.01 (rev 0)  
[CoppeliaSim:info] Model loaded.

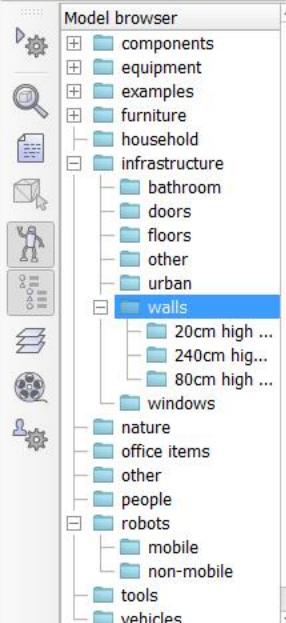
Input Lua code here, or type "help()" (use TAB for auto-completion)

Sandbox script

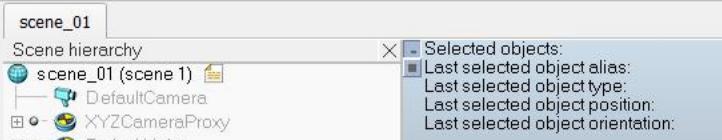
14



Bullet 2.78

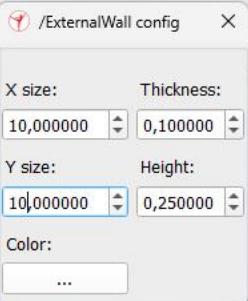
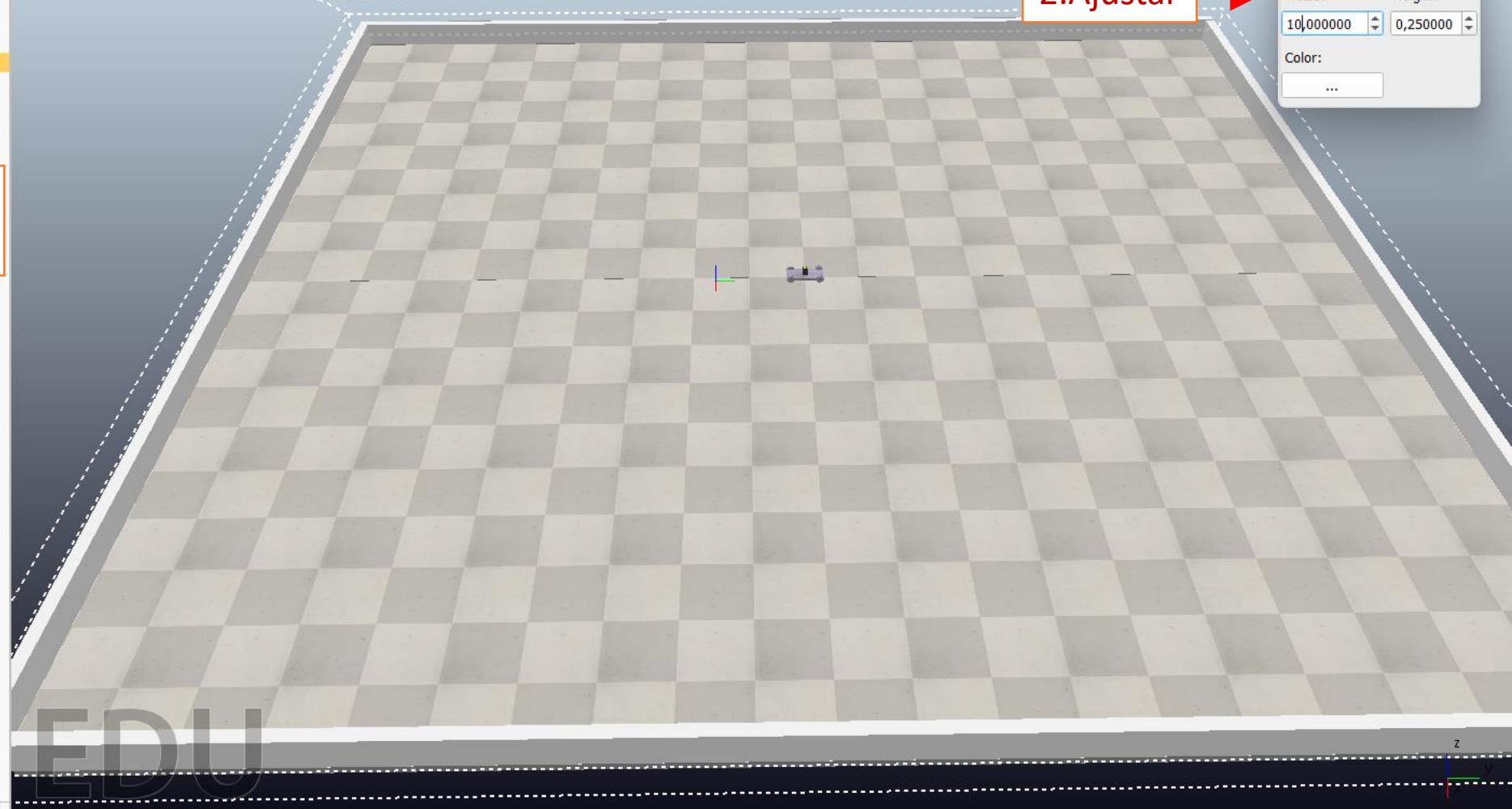


1. Doble click  
en icono



Selected objects:  
Last selected object alias:  
Last selected object type:  
Last selected object position:  
Last selected object orientation:

1 /ExternalWall (deprecated name:ExternalWall)  
Dummy  
x: +0.050 y: -0.800 z: 0.000  
a: 0.00 b: 0.00 g: 0.00



[CoppeliaSim:info] File was previously written with CoppeliaSim version 4.05.01 (rev 0)  
[CoppeliaSim:info] Model loaded.

Input Lua code here, or type "help()" (use TAB for auto-completion)

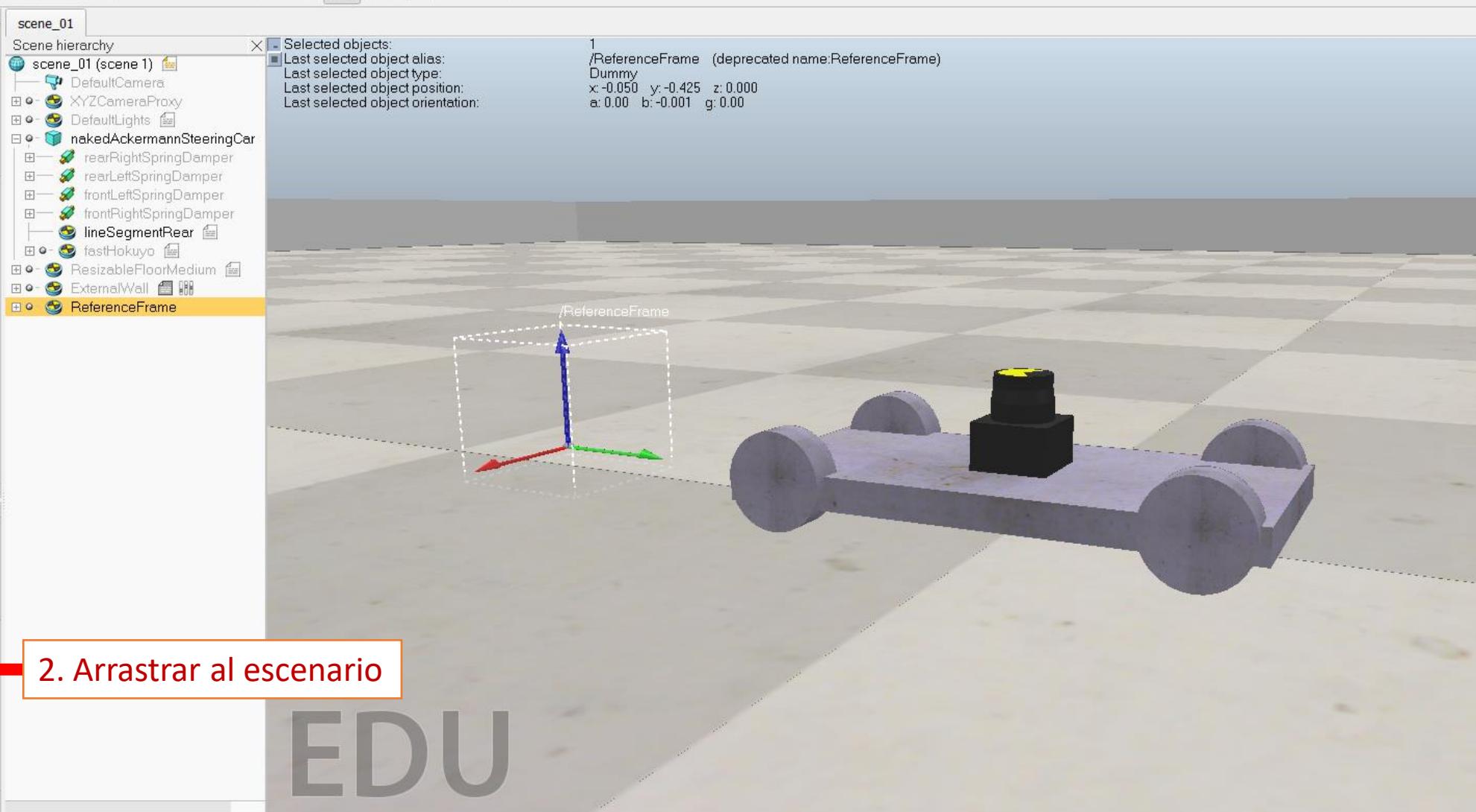
Sandbox script

# Sensor de pose en V-Rep



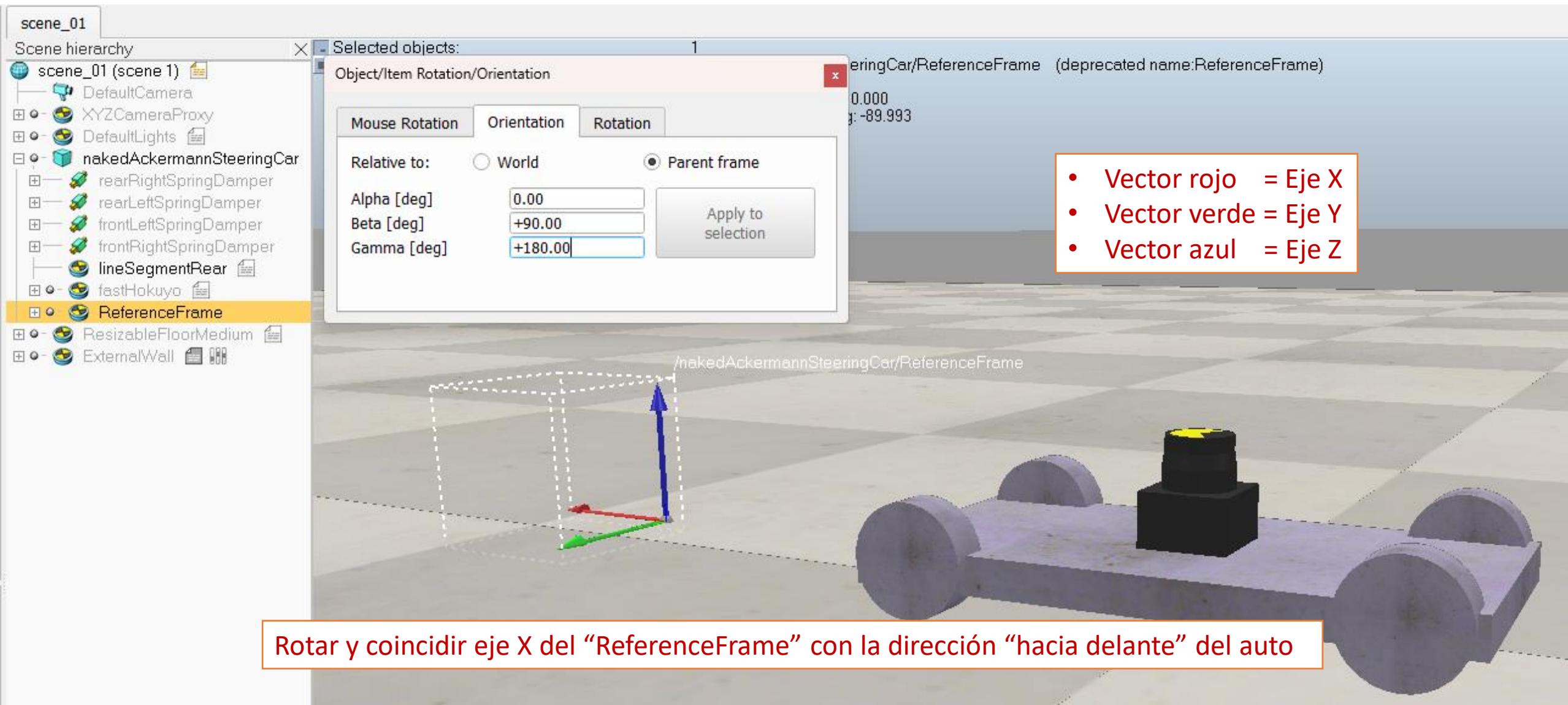
**Model browser**

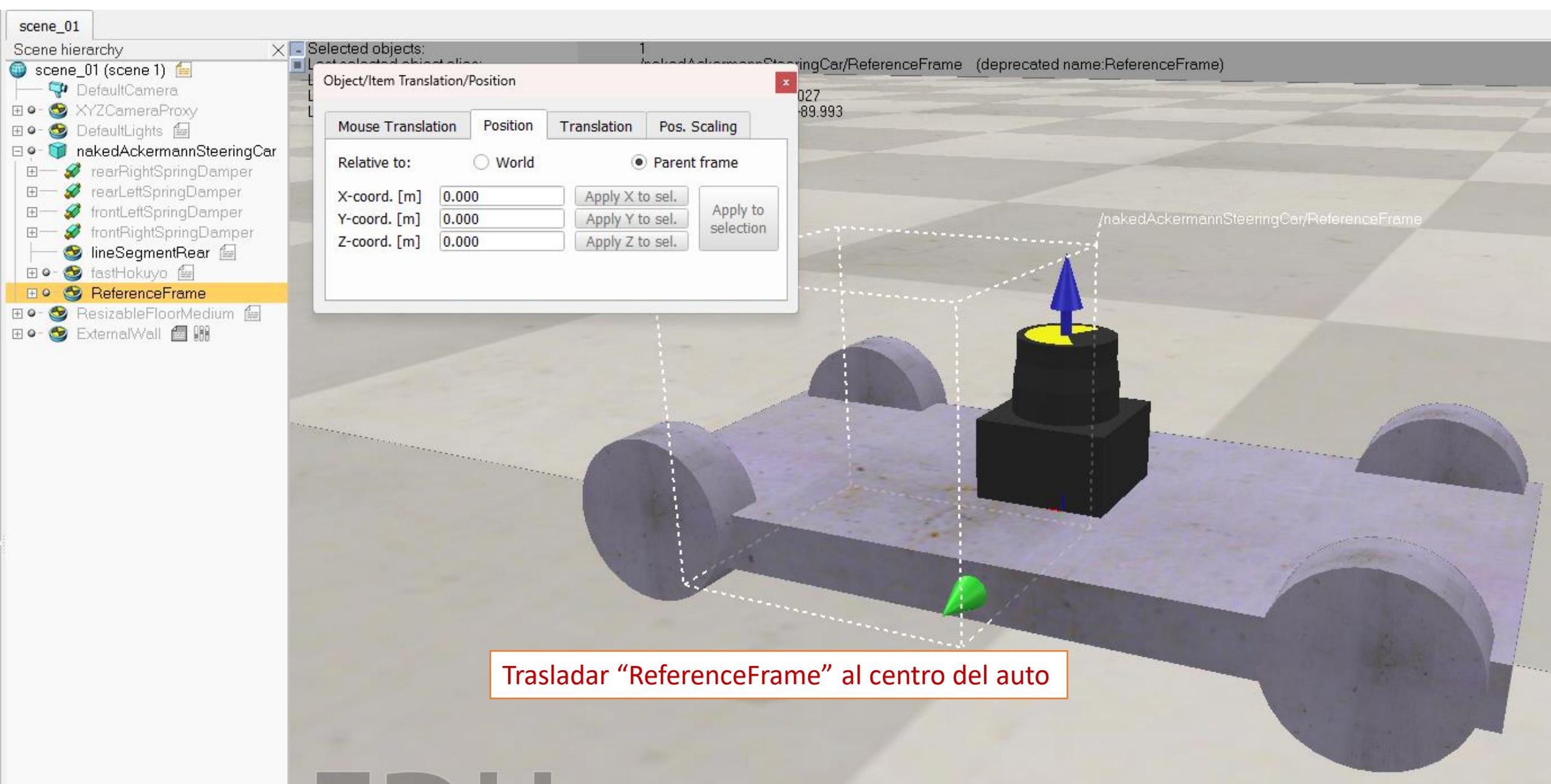
- plants
- shelves-cupboards-racks
- tables
- household
- infrastructure
  - bathroom
  - doors
  - floors
  - other
  - urban
- walls
  - 20cm high walls
  - 240cm high walls
  - 80cm high walls
- windows
- nature
- office items
- other
  - people
  - robots
  - mobile
  - non-mobile
  - tools
  - vehicles

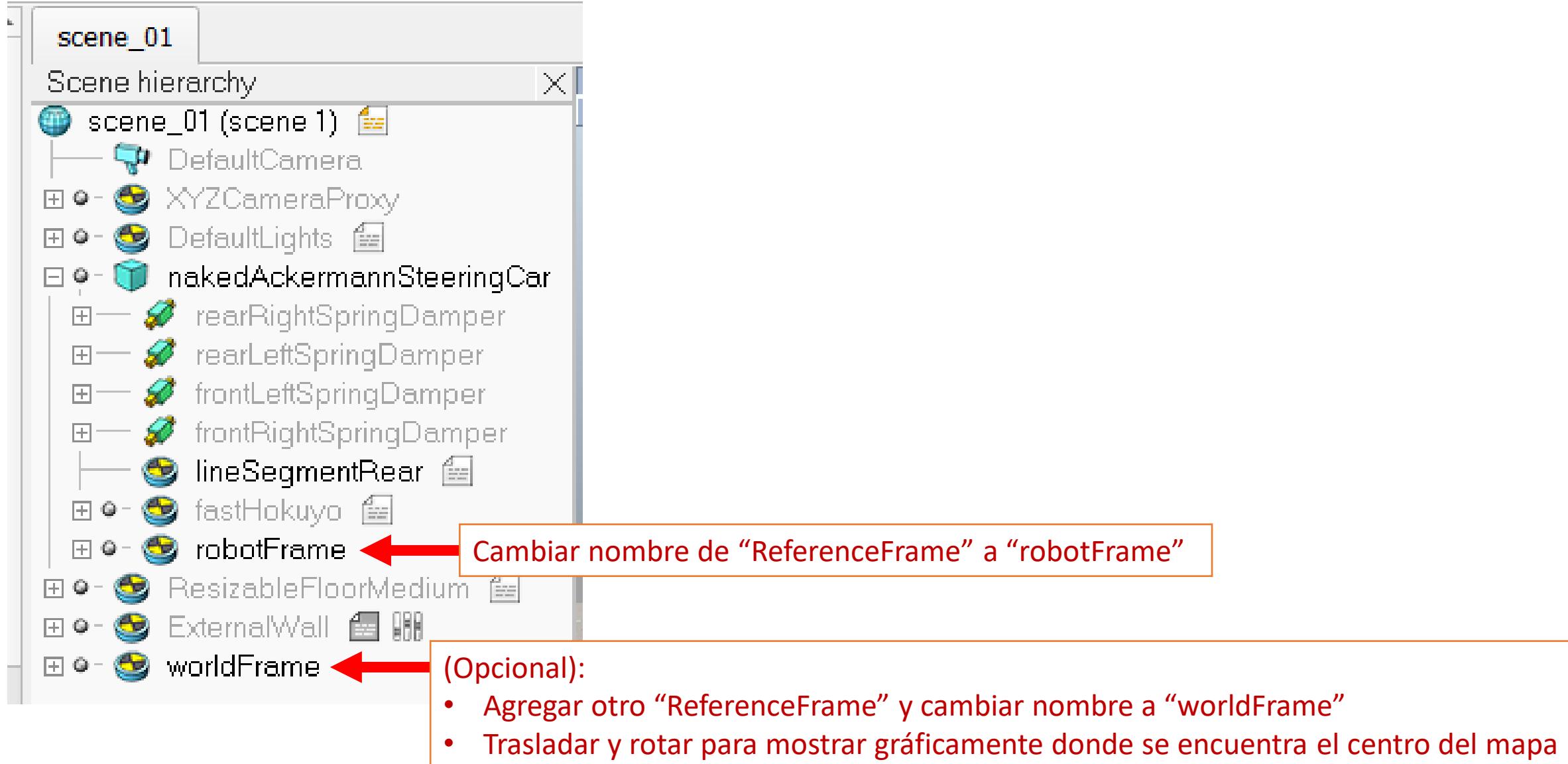


[sandboxScript:info] simulation stopping...  
[sandboxScript:info] Simulation stopped.  
[CoppeliaSim:info] Deleting selection...  
[CoppeliaSim:info] done.  
[CoppeliaSim:info] Loading model (C:/Program Files/CoppeliaRobotics/CoppeliaSimEdu/models/other/reference frame.ttm). Serialization version is 24.  
[CoppeliaSim:info] File was previously written with CoppeliaSim version 4.05.01 (rev 0)  
[CoppeliaSim:info] Model loaded.

Input Lua code here, or type "help()" (use TAB for auto-completion)

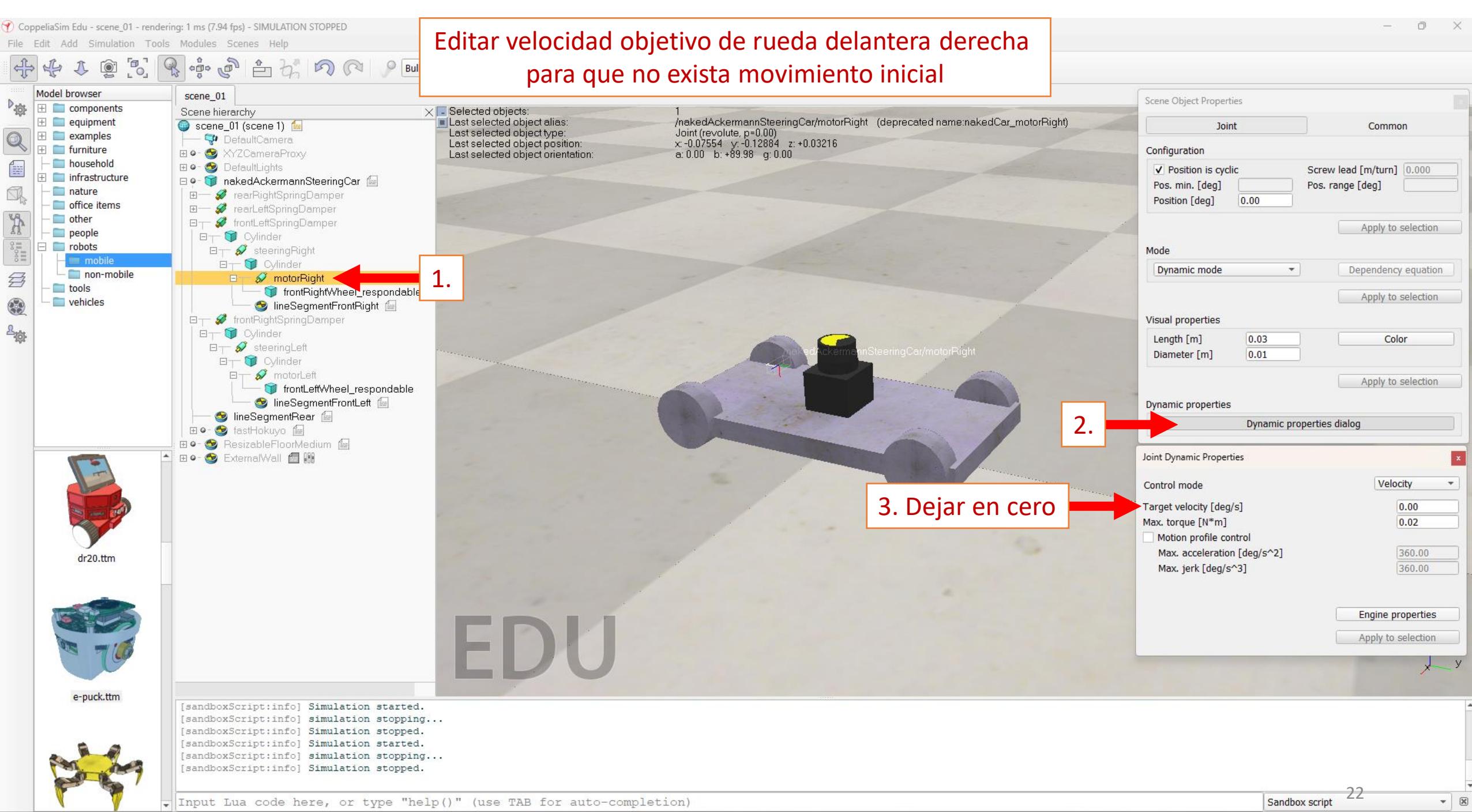






# Comunicación V-Rep -> Matlab

Teclado



CoppeliaSim Edu - scene\_01 - rendering: 2 ms (8.0 fps) - SIMULATION STOPPED

File Edit Add Simulation Tools Modules Scenes Help

Bullet 2.78

Model browser

- components
- equipment
- examples
- furniture
- household
- infrastructure
- nature
- office items
- other
- people
- robots
  - mobile
  - non-mobile
- tools
- vehicles

scene hierarchy

- scene\_01 (scene 1)
  - DefaultCamera
  - XYZCameraProxy
  - DefaultLights
  - nakedAckermannSteeringCar
    - rearRightSpringDamper
    - rearLeftSpringDamper
    - frontLeftSpringDamper
      - Cylinder
      - steeringRight
      - Cylinder
      - motorRight
      - frontRightWheel\_respondable
      - lineSegmentFrontRight
    - frontRightSpringDamper
      - Cylinder
      - steeringLeft
      - Cylinder
      - motorLeft
      - frontLeftWheel\_respondable
      - lineSegmentFrontLeft
    - lineSegmentRear
  - fastHokuyo
  - ResizableFloorMedium
  - ExternalWall

Selected objects:

- Last selected object alias: nakedAckermannSteeringCar/motorLeft (deprecated name:nakedCar\_motorLeft)
- Last selected object type: Joint (revolute, p:0.00)
- Last selected object position: x: +0.0755 y: -0.12884 z: +0.03216
- Last selected object orientation: a: 0.00 b: +89.98 g: 0.00

1.

2.

3. Dejar en cero

Scene Object Properties

Joint

Configuration

- Position is cyclic
- Screw lead [m/turn] 0.000
- Pos. min. [deg]
- Pos. range [deg]
- Position [deg] 0.00

Common

Apply to selection

Mode

- Dynamic mode
- Dependency equation

Visual properties

- Length [m] 0.03
- Diameter [m] 0.01
- Color

Apply to selection

Dynamic properties

Dynamic properties dialog

Joint Dynamic Properties

Control mode

- Velocity
- Target velocity [deg/s] 0.00
- Max. torque [N\*m] 0.02
- Motion profile control
- Max. acceleration [deg/s^2] 360.00
- Max. jerk [deg/s^3] 360.00

Engine properties

Apply to selection

EDU

[sandboxScript:info] Simulation started.  
[sandboxScript:info] simulation stopping...  
[sandboxScript:info] Simulation stopped.  
[sandboxScript:info] Simulation started.  
[sandboxScript:info] simulation stopping...  
[sandboxScript:info] Simulation stopped.

Input Lua code here, or type "help()" (use TAB for auto-completion)

Sandbox script

23

Model browser

- components
- equipment
- examples
- furniture
- household
- infrastructure
- nature
- office items
- other
- people
- robots
  - mobile
  - non-mobile
- tools
- vehicles

scene\_01

Scene hierarchy

- scene\_01 (scene 1)
  - DefaultCamera
  - XYZCameraProxy
  - DefaultLights
  - nakedAckermannSteeringCar
  - ResizableFloorMedium
  - ExternalWall

Selected objects:

- Last selected object alias: nakedAckermannSteeringCar (deprecated name:nakedAckermannSteeringCar)
- Last selected object type: Shape (cuboid)
- Last selected object position: x: 0.000 y: 0.000 z: +0.027
- Last selected object orientation: a: +90.386 b: +0.007 g: +90.003

Scripts

Main script

Child script "/nakedAckermannSteeringCar"

Child script "/nakedAckermannSteeringCar/lineSegmentRear"

Child script "/nakedAckermannSteeringCar/lineSegmentFrontRight"

Child script "/nakedAckermannSteeringCar/lineSegmentFrontLeft"

Child script "/nakedAckermannSteeringCar/ ... /sensor1"

Child script "/nakedAckermannSteeringCar/ ... /sensor2"

Child script "/nakedAckermannSteeringCar/fastHokuyo"

Child script "/ResizableFloorMedium"

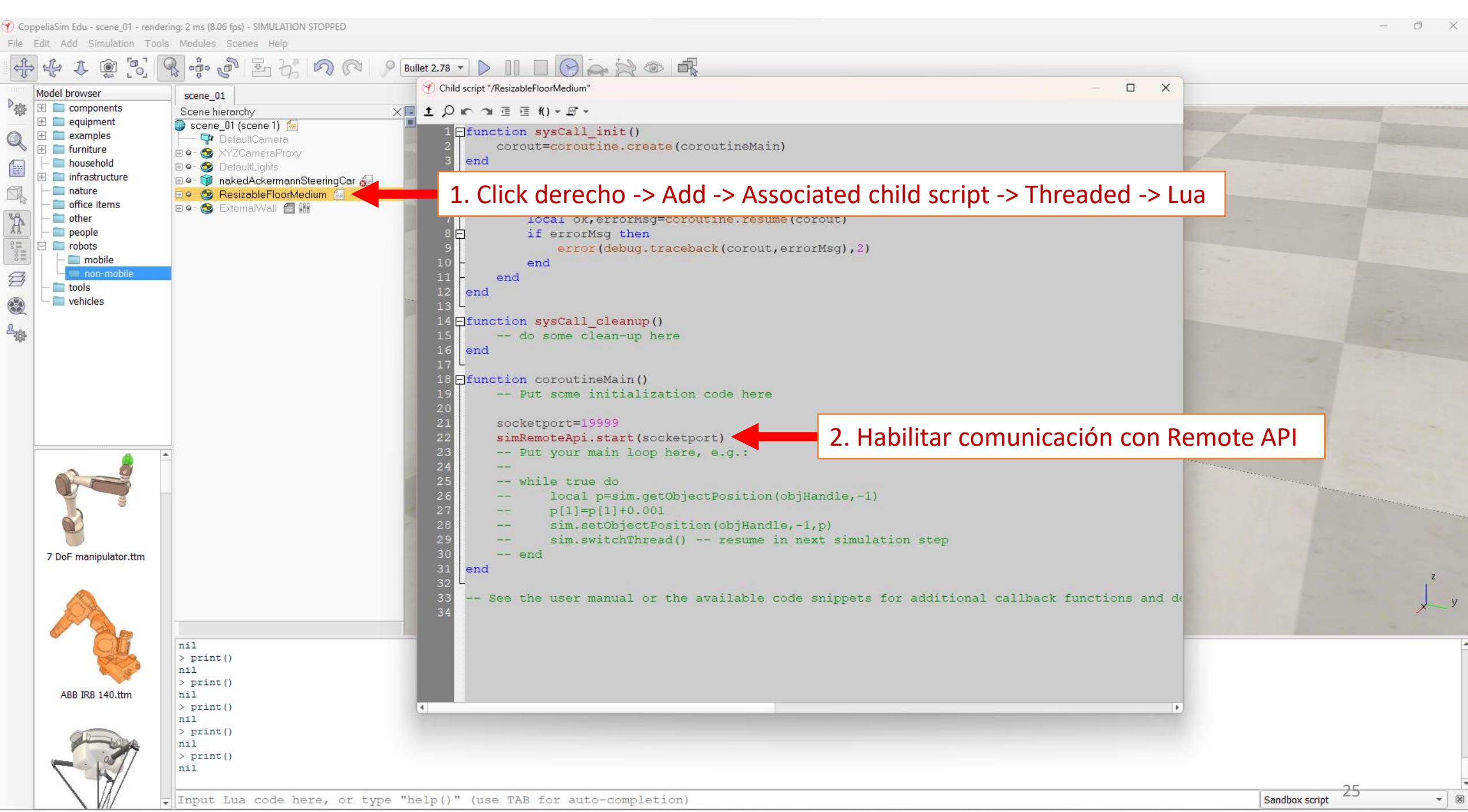
Script properties

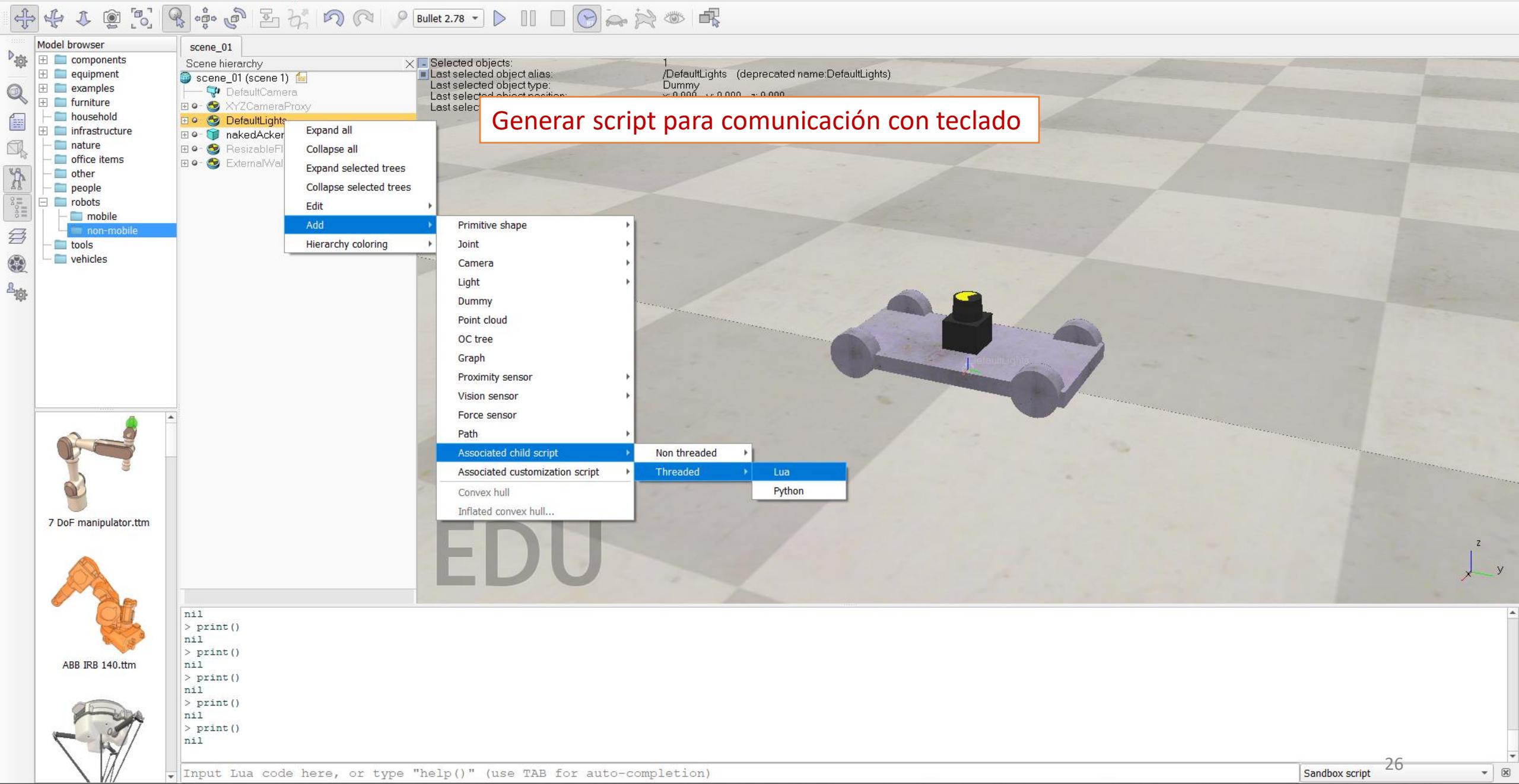
Disabled

Execution priority: normal

Deshabilitar script por defecto del auto

The screenshot shows the CoppeliaSim Edu software interface. On the left, there's a 'Model browser' with categories like components, equipment, and robots. Below it are icons for various models: '7 DoF manipulator.ttm', 'ABB IRB 140.ttm', and a camera icon. The main workspace shows a 3D scene with a car model and a large 'EDU' watermark. A red arrow points from the text 'Deshabilitar script por defecto del auto' to the 'Disabled' checkbox in the 'Script properties' panel. Another red arrow points to the 'nakedAckermannSteeringCar' entry in the 'Child script' list. The top status bar indicates the simulation is stopped.





## Generar script para comunicación con teclado



- Regular API reference

<https://www.coppeliarobotics.com/helpFiles/index.html>

## Messaging

```
sim.addLog
sim.loadModule
sim.unloadModule
sim.getSimulatorMessage
sim.serialCheck
sim.serialClose
sim.serialOpen
sim.serialRead
sim.serialSend
sim.setInt32Signal
sim.getInt32Signal
sim.clearInt32Signal
sim.setFloatSignal
sim.getFloatSignal
sim.clearFloatSignal
sim.setStringSignal
sim.getStringSignal
sim.clearStringSignal
sim.getSignalName
sim.waitForSignal
sim.assistantConsoleClose
sim.assistantConsoleOpen
sim.assistantConsolePrint
sim.persistentDataRead
sim.persistentDataWrite
sim.getPersistentDataTags
sim.getMatchingPersistent
sim.callScriptFunction
sim.getScriptFunctions
sim.executeScriptString
sim.getObjectUid
sim.getGenesisEvents
sim.broadcastMsg
sim.pushUserEvent
```

CoppeliaSim script

# Regular API function

## simSetInt32Signal / sim.setInt32Signal

Description	Sets the value of an integer signal. A signal created in a <a href="#">child script</a> , a <a href="#">customization script</a> or in the <a href="#">main script</a> will be automatically cleared when the script ends. See also the <a href="#">other signal functions</a> .
C/C++ synopsis	<code>int simSetInt32Signal(const char* signalName,int signalValue)</code>
C/C++ parameters	<b>signalName</b> : name of the signal <b>signalValue</b> : value of the signal
C/C++ return value	-1 if operation was not successful
Lua synopsis	<code>sim.setInt32Signal(string signalName,int signalValue)</code>
Lua parameters	Similar to the C-function counterpart
Lua return values	
Python synopsis	<code>sim.setInt32Signal(string signalName,int signalValue)</code>

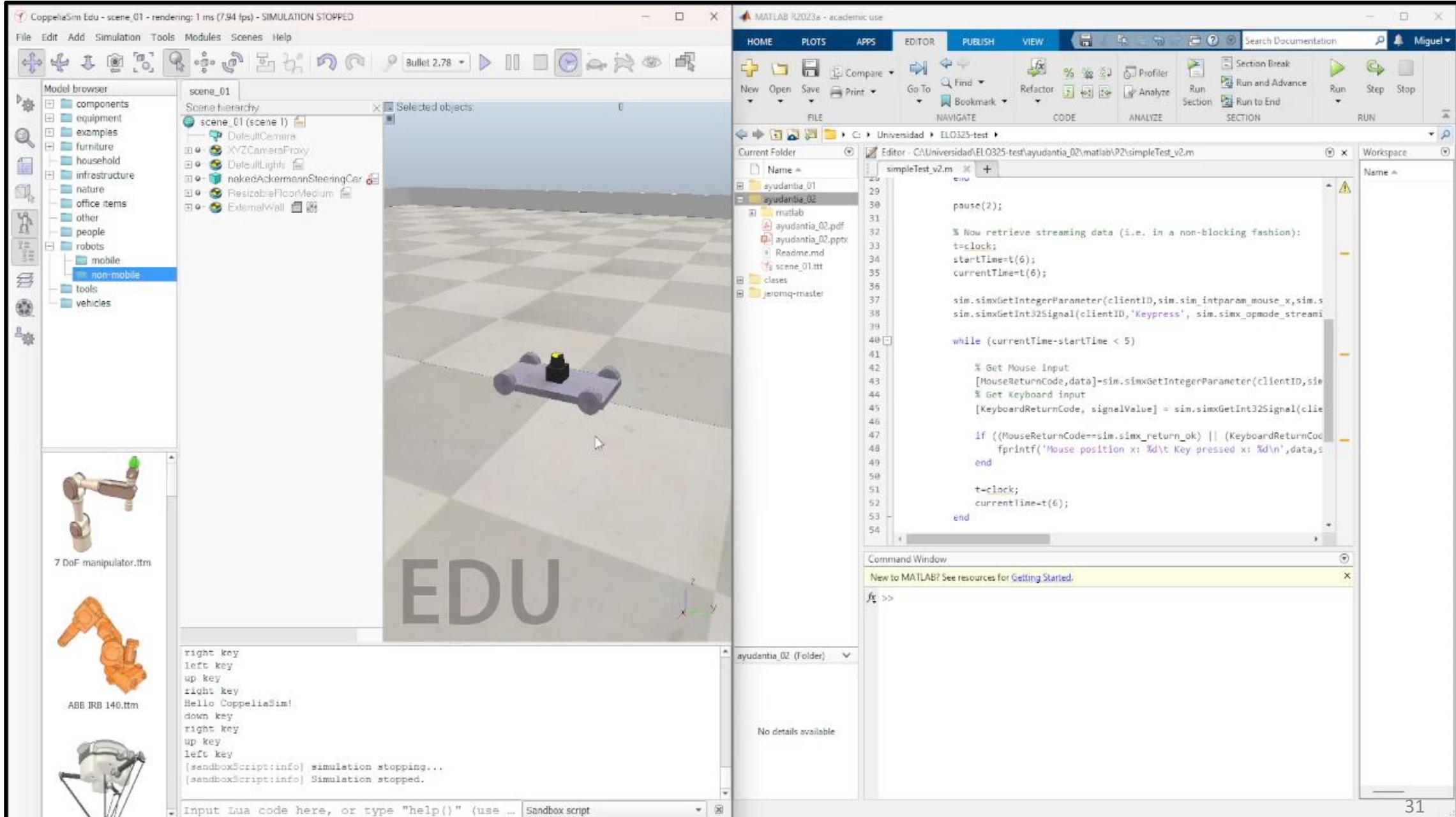
- Legacy Remote API (MATLAB)

<https://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsMatlab.htm#simxGetInt32Param>

simxGetInt32Signal	
Description	Gets the value of an integer signal. Signals are cleared at simulation start. See also <a href="#">simxSetInt32Signal</a> , <a href="#">simxClearInt32Signal</a> , <a href="#">simxGetFloatSignal</a> and <a href="#">simxGetStringSignal</a> .
Matlab synopsis	[number returnCode,number signalValue]=simx <b>GetInt32Signal</b> (number clientID,string signalName,number operationMode)
Matlab parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>signalName</b> : name of the signal <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <a href="#">simx_opmode_streaming</a> (the first call) and <a href="#">simx_opmode_buffer</a> (the following calls)
Matlab return values	<b>returnCode</b> : a <a href="#">remote API function return code</a> <b>signalValue</b> : the value of the signal
Other languages	<a href="#">C/C++</a> , <a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Octave</a>

```
17     clientID=sim.simxStart('127.0.0.1',19999,true,true,5000,5);
18
19     if (clientID>-1)
20         disp('Connected to remote API server');
21
22     % Now try to retrieve data in a blocking fashion (i.e. a service call):
23     [res,objs]=sim.simxGetObjects(clientID,sim.sim_handle_all,sim.simx_opmode_blocking);
24     if (res==sim.simx_return_ok)
25         fprintf('Number of objects in the scene: %d\n',length(objs));
26     else
27         fprintf('Remote API function call returned with error code: %d\n',res);
28     end
29
30     pause(2);
31
32     % Now retrieve streaming data (i.e. in a non-blocking fashion):
33     t=clock;
34     startTime=t(6);
35     currentTime=t(6);
36
37     sim.simxGetIntegerParameter(clientID,sim.sim_intparam_mouse_x,sim.simx_opmode_streaming); % Initialize streaming
38     sim.simxGetInt32Signal(clientID,'Keypress', sim.simx_opmode_streaming); % Initialize streaming
39
40     while (currentTime-startTime < 5)
41
42         % Get Mouse input
43         [MouseReturnCode,data]=sim.simxGetIntegerParameter(clientID,sim.sim_intparam_mouse_x,sim.simx_opmode_buffer); % Try to retrieve the streamed data
44         % Get Keyboard input
45         [KeyboardReturnCode, signalValue] = sim.simxGetInt32Signal(clientID,'Keypress', sim.simx_opmode_buffer);
46
47         if ((MouseReturnCode==sim.simx_return_ok) || (KeyboardReturnCode==sim.simx_return_ok)) % After initialization of streaming, it will take a few ms before
48             fprintf('Mouse position x: %d\t Key pressed x: %d\n',data,signalValue); % Mouse position x, and Keyboard is actualized when the cursor is over Coppel
49         end
50
51         t=clock;
52         currentTime=t(6);
53     end
54
55     % Now send some data to CoppeliaSim in a non-blocking fashion:
```

# Video



# Comunicación Matlab -> V-Rep

Teclado

```
pioneerExample.m * simpleTest_v2p1.m * +  
42  
43 while (currentTime-startTime < 50)  
44  
45 % Get Keyboard input  
46 [KeyboardReturnCode, signalValue] = sim.simxGetInt32Signal(clientID,'Keypress', sim.simx_opmode_buffer); }  
47  
48 if (KeyboardReturnCode==sim.simx_return_ok) % After initialization of streaming, it will take a few ms before the first value arrives, so check the return code  
49 if (signalValue==2007)  
50 %-- up key  
51 desiredWheelRotSpeed = desiredWheelRotSpeed + wheelRotSpeedDx;  
52 fprintf('Up key,\t RotSpeed = %d\n',desiredWheelRotSpeed*180/pi);  
53 end  
54 if (signalValue==2008)  
55 %-- down key  
56 desiredWheelRotSpeed = desiredWheelRotSpeed - wheelRotSpeedDx;  
57 fprintf('Down key,\t RotSpeed = %d\n',desiredWheelRotSpeed*180/pi);  
58 end  
59 if (signalValue==2009)  
60 %-- left key  
61 desiredSteeringAngle = desiredSteeringAngle + steeringAngleDx;  
62 if (desiredSteeringAngle>45*pi/180)  
63 desiredSteeringAngle=45*pi/180;  
64 end  
65 fprintf('Left key,\t SteerAngle = %d\n',desiredSteeringAngle*180/pi);  
66 end  
67 if (signalValue==2010)  
68 %-- right key  
69 desiredSteeringAngle = desiredSteeringAngle - steeringAngleDx;  
70 if (desiredSteeringAngle<-45*pi/180)  
71 desiredSteeringAngle=-45*pi/180;  
72 end  
73 fprintf('Right key,\t SteerAngle = %d\n',desiredSteeringAngle*180/pi);  
74 end  
75 end  
76  
77 %% We handle the front left and right wheel steerings (Ackermann steering):  
78 steeringAngleLeft = atan(1/(-d+l/tan(desiredSteeringAngle)));  
79 steeringAngleRight= atan(1/(d+l/tan(desiredSteeringAngle)));  
80 [~,~] = sim.simxSetJointTargetPosition(clientID, steeringLeft, steeringAngleLeft, sim.simx_opmode_streaming); }  
81 [~,~] = sim.simxSetJointTargetPosition(clientID, steeringRight, steeringAngleRight, sim.simx_opmode_streaming); }  
82
```

Lectura de señal Int32 desde V-Rep

Chequeo de tecla presionada

Seteo ángulos ruedas delanteras

```

74     end
75
76
77     %% We handle the front left and right wheel steerings (Ackermann steering):
78     steeringAngleLeft = atan(1/(-d+l/tan(desiredSteeringAngle)));
79     steeringAngleRight= atan(1/(d+l/tan(desiredSteeringAngle)));
80     [~] = sim.simxSetJointTargetPosition(clientID, steeringLeft, steeringAngleLeft, sim.simx_opmode_streaming);
81     [~] = sim.simxSetJointTargetPosition(clientID, steeringRight, steeringAngleRight, sim.simx_opmode_streaming);
82
83     %% We take care of setting the desired wheel rotation speed:
84     [~] = sim.simxSetJointTargetVelocity(clientID, motorLeft , desiredWheelRotSpeed , sim.simx_opmode_streaming);
85     [~] = sim.simxSetJointTargetVelocity(clientID, motorRight , desiredWheelRotSpeed , sim.simx_opmode_streaming);
86
87     pause(0.01) ] Control de actualización de datos (100 Hz) -> mejorable
88
89     t=clock;
90     currentTime=t(6);
91 end
92
93 % Stop motors
94 [~] = sim.simxSetJointTargetVelocity(clientID, motorLeft , 0 , sim.simx_opmode_streaming);
95 [~] = sim.simxSetJointTargetVelocity(clientID, motorRight , 0 , sim.simx_opmode_streaming);
96
97 % Now send some data to CoppeliaSim in a non-blocking fashion:
98 sim.simxAddStatusbarMessage(clientID,'Hello CoppeliaSim!',sim.simx_opmode_oneshot);
99
100 % Before closing the connection to CoppeliaSim, make sure that the last command sent out had time to arrive. You can guarantee this with (for example):
101 sim.simxGetPingTime(clientID);
102
103 % Now close the connection to CoppeliaSim:
104 sim.simxFinish(clientID);
105 else
106     disp('Failed connecting to remote API server');
107 end
108 sim.delete(); % call the destructor!
109
110 disp('Program ended');
111
112 end

```

}] Seteo ángulos ruedas del.

}] Seteo velocidad angular de ruedas del.

}] Cierra comunicación con V-Rep

## simxSetJointTargetPosition

Description	Sets the target position of a joint if the joint is in torque/force mode (also make sure that the joint's motor and position control are enabled). See also <a href="#">simxSetJointPosition</a> .
Matlab synopsis	[number returnCode]= <a href="#">simxSetJointTargetPosition</a> (number clientID,number jointHandle,number targetPosition,number operationMode)
Matlab parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>jointHandle</b> : handle of the joint <b>targetPosition</b> : target position of the joint (angular or linear value depending on the joint type) <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <a href="#">simx_opmode_oneshot</a> or <a href="#">simx_opmode_streaming</a>
Matlab return values	<b>returnCode</b> : a <a href="#">remote API function return code</a>
Other languages	<a href="#">C/C++</a> , <a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Octave</a>

## simxSetJointTargetVelocity

Description	Sets the intrinsic target velocity of a non-spherical joint. This command makes only sense when the joint mode is in torque/force mode: the dynamics functionality and the joint motor have to be enabled (position control should however be disabled)
Matlab synopsis	[number returnCode]= <a href="#">simxSetJointTargetVelocity</a> (number clientID,number jointHandle,number targetVelocity,number operationMode)
Matlab parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>jointHandle</b> : handle of the joint <b>targetVelocity</b> : target velocity of the joint (linear or angular velocity depending on the joint-type) <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <a href="#">simx_opmode_oneshot</a> or <a href="#">simx_opmode_streaming</a>
Matlab return values	<b>returnCode</b> : a <a href="#">remote API function return code</a>
Other languages	<a href="#">C/C++</a> , <a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Octave</a>

# Comunicación V-Rep -> Matlab

Sensor Hokuyo

Bullet 2.78

Child script "/nakedAckermannSteeringCar/fastHokuyo"

Scene hierarchy

- scene\_01
  - DefaultCamera
  - XYZCameraProxy
  - DefaultLights
  - nakedAckermannSteeringCar
    - rearRightSpringDamper
    - rearLeftSpringDamper
    - frontLeftSpringDamper
    - frontRightSpringDamper
    - lineSegmentRear
    - fastHokuyo
  - ResizableFloorMedium
  - ExternalWall

Editar script por defecto del sensor Hokuyo

```
for j=0,u[2]-1,1 do
    for k=0,u[1]-1,1 do
        local w=2+4*(j*u[1]+k)
        local v={u[w+1],u[w+2],u[w+3],u[w+4]}
        if generateData then
            if rangeData then
                table.insert(measuredData,v[4])
            else
                if v[4]<maxScanDistance*0.9999 or not discardMaxDistPts then
                    p=sim.multiplyVector(relRefM,v)
                    table.insert(measuredData,p[1])
                    table.insert(measuredData,p[2])
                    table.insert(measuredData,p[3])
                end
            end
        end
        if showLines then
            p=sim.multiplyVector(sensorM,v)
            t[4]=p[1]
            t[5]=p[2]
            t[6]=p[3]
            sim.addDrawingObjectItem(lines,t)
        end
    end
end
--print(type(measuredData))
--print(table.concat(measuredData,","))
--print("length of measuredData: ", #measuredData)

local packedData=sim.packFloatTable(measuredData,0,0)
sim.setStringSignal('measuredData',packedData)
```

Convierte tabla en string y envía señal a Remote API

37

- Regular API reference

<https://www.coppeliarobotics.com/helpFiles/index.html>

## Packing / unpacking

```
sim.packTable
sim.unpackTable
sim.packUInt8Table
sim.unpackUInt8Table
sim.packUInt16Table
sim.unpackUInt16Table
sim.packUInt32Table
sim.unpackUInt32Table
sim.packInt32Table
sim.unpackInt32Table
sim.packFloatTable
sim.unpackFloatTable
sim.packDoubleTable
sim.unpackDoubleTable
sim.copyTable
sim.transformBuffer
```

CoppeliaSim script

### simPackFloatTable / sim.packFloatTable

Description	Packs a table of floating-point numbers into a string. See also <a href="#">sim.unpackFloatTable</a> and the <a href="#">other packing/unpacking functions</a> .
C/C++ synopsis	
C/C++ parameters	
C/C++ return value	
Lua synopsis	buffer data=sim.packFloatTable(float[] floatingNumbers,int startFloatIndex=0,int floatCount=0)
Lua parameters	<b>floatingNumbers:</b> a table containing floating-point numbers. Non-numbers will be packed as zero values <b>startFloatIndex:</b> the zero-based index from which on data should be packed. Can be omitted in which case 0 is used <b>floatCount:</b> the amount of floats that should be packed. Can be omitted in which case 0 is used (which indicates that the maximum available floats should be packed from the indicated startFloatIndex)
Lua return values	<b>data:</b> a string (values between 0 and 255) that contains packed floating-point numbers from the table
Python synopsis	bytes data=sim.packFloatTable(list floatingNumbers,int startFloatIndex=0,int floatCount=0)

- Regular API reference

<https://www.coppeliarobotics.com/helpFiles/index.html>

## Messaging

```
sim.addLog
sim.loadModule
sim.unloadModule
sim.getSimulatorMessage
sim.serialCheck
sim.serialClose
sim.serialOpen
sim.serialRead
sim.serialSend
sim.setInt32Signal
sim.getInt32Signal
sim.clearInt32Signal
sim.setFloatSignal
sim.getFloatSignal
sim.clearFloatSignal
sim.setStringSignal
sim.getStringSignal
sim.clearStringSignal
sim.getSignalName
sim.waitForSignal
sim.assistantConsoleClose
sim.assistantConsoleOpen
sim.assistantConsolePrint
sim.persistentDataRead
sim.persistentDataWrite
sim.getPersistentDataTags
sim.getMatchingPersistentDataTags
sim.callScriptFunction
sim.getScriptFunctions
```

CoppeliaSim script

### simSetStringSignal / sim.setStringSignal

Description	Sets the value of a string signal. A signal created in a <b>child script</b> , a <b>customization script</b> or in the <b>main script</b> will be automatically cleared when the script ends. See also the <b>other signal functions</b> .
C/C++ synopsis	int simSetStringSignal(const char* signalName,const char* signalValue,int stringLength)
C/C++ parameters	<b>signalName</b> : name of the signal <b>signalValue</b> : value of the signal (which may contain any value, including embedded zeros) <b>stringLength</b> : the size of the string value.
C/C++ return value	-1 if operation was not successful
Lua synopsis	sim.setStringSignal(string signalName,buffer signalValue)
Lua parameters	Similar to the C-function counterpart
Lua return values	
Python synopsis	sim.setStringSignal(string signalName,string/bytes signalValue)

- Legacy Remote API (MATLAB)

<https://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsMatlab.htm#simxGetStringSignal>

simxGetStringSignal	
Description	Gets the value of a string signal. Signals are cleared at simulation start. To pack/unpack integers/floats into/from a string, refer to <a href="#">simxPackInts</a> , <a href="#">simxPackFloats</a> , <a href="#">simxUnpackInts</a> and <a href="#">simxUnpackFloats</a> . See also <a href="#">simxSetStringSignal</a> , <a href="#">simxReadStringStream</a> , <a href="#">simxClearStringSignal</a> , <a href="#">simxGetInt32Signal</a> and <a href="#">simxGetFloatSignal</a> .
Matlab synopsis	[number returnCode,string signalValue]= <a href="#">simxGetStringSignal</a> (number clientID,string signalName,number operationMode)
Matlab parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>signalName</b> : name of the signal <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <a href="#">simx_opmode_streaming</a> (the first call) and <a href="#">simx_opmode_buffer</a> (the following calls)
Matlab return values	<b>returnCode</b> : a <a href="#">remote API function return code</a> <b>signalValue</b> : the signal data (that may contain any value, including embedded zeros).
Other languages	<a href="#">C/C++</a> , <a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Octave</a>

```
1 function simpleTest()
2     disp('Program started');
3     sim=remApi('remoteApi'); % using the prototype file (remoteApiProto.m)
4     sim.simxFinish(-1); % just in case, close all opened connections
5     clientID=sim.simxStart('127.0.0.1',19999,true,true,5000,5);
6
7     if (clientID>-1)
8         disp('Connected to remote API server');
9
10    % Now retrieve streaming data (i.e. in a non-blocking fashion):
11    t=clock;
12    startTime=t(6);
13    currentTime=t(6);
14
15    % Initialize streaming
16    sim.simxGetInt32Signal(clientID,'Keypress',sim.simx_opmode_streaming);
17    sim.simxGetStringSignal(clientID,'measuredData',sim.simx_opmode_streaming);
18
19    [~, steeringLeft] = sim.simxGetObjectHandle(clientID, './steeringLeft', sim.simx_opmode_blocking);
20    [~, steeringRight] = sim.simxGetObjectHandle(clientID, './steeringRight', sim.simx_opmode_blocking);
21    [~, motorLeft] = sim.simxGetObjectHandle(clientID, './motorLeft', sim.simx_opmode_blocking);
22    [~, motorRight] = sim.simxGetObjectHandle(clientID, './motorRight', sim.simx_opmode_blocking);
23    desiredSteeringAngle = 0;
24    desiredWheelRotSpeed = 0;
25    steeringAngleDx = 2*pi/180;
26    wheelRotSpeedDx = 20*pi/180;
27    d=0.755; %-- 2*d=distance between left and right wheels
28    l=2.5772; %-- l=distance between front and rear wheels
29
```

```
pioneerExample.m simpleTest_v2p1.m +  
24     desiredSteeringAngle = 0;  
25     desiredWheelRotSpeed = 0;  
26     steeringAngleDx      = 2*pi/180;  
27     wheelRotSpeedDx      = 20*pi/180;  
28     d=0.755; %-- 2*d=distance between left and right wheels  
29     l=2.5772; %-- l=distance between front and rear wheels  
30  
31     figure(1)  
32     degrees = linspace(-120,120,684);  
33     radians = degrees.*pi/180;  
34     xlabel('Angle (°)'); ylabel('Distance [m]');  
35     grid on;  
36     xlim([-120,120]);  
37     ylim([0,10]);  
38  
39 while (currentTime-startTime < 50)  
40  
41     [err,signal]=sim.simxGetStringSignal(clientID,'measuredData',sim.simx_opmode_buffer);  
42     if (err==sim.simx_return_ok)  
43         % Data produced by the child script was retrieved!  
44         unpackedsignal = sim.simxUnpackFloats(signal);  
45         % disp(unpackedsignal)|  
46         p = polarscatter(radians,unpackedsignal);  
47         p.Marker = ".";  
48         drawnow;  
49     end  
50  
51     % Get Keyboard input  
52     [KeyboardReturnCode, signalValue1] = sim.simxGetIntegerSignal(clientID, 'Keyboard');
```

Preámbulo para gráfico Polar.

Datasheet Hokuyo:

- Ángulo de visión = 240°
- N° de puntos = 684

Captura de datos del sensor y actualización de gráfico Polar

# Comunicación con Matlab

Sensor de pose del robot

## simxGetObjectPosition

Description	Retrieves the position of an object. See also <a href="#">simxSetObjectPosition</a> , <a href="#">simxGetObjectOrientation</a> , <a href="#">simxGetObjectQuaternion</a> and <a href="#">simxGetObjectGroupData</a> .
Matlab synopsis	[number returnCode,array position]=simxGetObjectPosition(number clientID,number objectHandle,number relativeToObjectHandle,number operationMode)
Matlab parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>objectHandle</b> : handle of the object <b>relativeToObjectHandle</b> : indicates relative to which reference frame we want the position. Specify -1 to retrieve the absolute position, sim_handle_parent to retrieve the position relative to the object's parent, or an object handle relative to whose reference frame you want the position <b>operationMode</b> : a remote API function operation mode. Recommended operation modes for this function are <a href="#">simx_opmode_streaming</a> (the first call) and <a href="#">simx_opmode_buffer</a> (the following calls)
Matlab return values	<b>returnCode</b> : a remote API function return code <b>position</b> : 3 values representing the position
Other languages	<a href="#">C/C++</a> , <a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Octave</a>

## simxGetObjectOrientation

Description	Retrieves the orientation (Euler angles) of an object. See also <a href="#">simxSetObjectOrientation</a> , <a href="#">simxGetObjectQuaternion</a> , <a href="#">simxGetObjectPosition</a> and <a href="#">simxGetObjectGroupData</a> .
Matlab synopsis	[number returnCode,array eulerAngles]=simxGetObjectOrientation(number clientID,number objectHandle,number relativeToObjectHandle,number operationMode)
Matlab parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>objectHandle</b> : handle of the object <b>relativeToObjectHandle</b> : indicates relative to which reference frame we want the orientation. Specify -1 to retrieve the absolute orientation, sim_handle_parent to retrieve the orientation relative to the object's parent, or an object handle relative to whose reference frame you want the orientation <b>operationMode</b> : a remote API function operation mode. Recommended operation modes for this function are <a href="#">simx_opmode_streaming</a> (the first call) and <a href="#">simx_opmode_buffer</a> (the following calls)
Matlab return values	<b>returnCode</b> : a remote API function return code <b>eulerAngles</b> : 3 values representing the Euler angles (alpha, beta and gamma)
Other languages	<a href="#">C/C++</a> , <a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Octave</a>

```
8     if (clientID>-1)
9         disp('Connected to remote API server');
10
11    % Now retrieve streaming data (i.e. in a non-blocking fashion):
12    t=clock;
13    startTime=t(6);
14    currentTime=t(6);
15
16    % Initialize streaming
17    sim.simxGetInt32Signal(clientID, 'Keypress', sim.simx_opmode_streaming);
18    sim.simxGetStringSignal(clientID, 'measuredData', sim.simx_opmode_streaming);
19
20    [~, refFrame] = sim.simxGetObjectHandle(clientID, './robotFrame', sim.simx_opmode_blocking);
21    sim.simxGetObjectPosition(clientID, refFrame, -1, sim.simx_opmode_streaming);
22    sim.simxGetObjectOrientation(clientID, refFrame, -1, sim.simx_opmode_streaming);
23
24    [~, steeringLeft] = sim.simxGetObjectHandle(clientID, './steeringLeft', sim.simx_opmode_blocking);
25    [~, steeringRight] = sim.simxGetObjectHandle(clientID, './steeringRight', sim.simx_opmode_blocking);
26    [~, motorLeft] = sim.simxGetObjectHandle(clientID, './motorLeft', sim.simx_opmode_blocking);
27    [~, motorRight] = sim.simxGetObjectHandle(clientID, './motorRight', sim.simx_opmode_blocking);
28    desiredSteeringAngle = 0;
29    desiredWheelRotSpeed = 0;
30    steeringAngleDx = 2*pi/180;
31    wheelRotSpeedDx = 20*pi/180;
32    d=0.755; %-- 2*d=distance between left and right wheels
33    l=2.5772; %-- l=distance between front and rear wheels
34
35    figure(1)
```

```
34
35     figure(1)
36     degrees = linspace(-120,120,684);
37     radians = degrees.*pi/180;
38     xlabel('Angle (°)'); ylabel('Distance [m]');
39     grid on;
40     xlim([-120,120]);
41     ylim([0,10]);
42
43 while (currentTime-startTime < 50)
44
45     [err, position] = sim.simxGetObjectPosition(clientID, refFrame, -1, sim.simx_opmode_buffer);
46     if(err==sim.simx_return_ok)
47         fprintf('(x,y,z)=(%.1f, %.1f, %.1f)\t',position);
48     end
49     [err, orientation] = sim.simxGetObjectOrientation(clientID, refFrame, -1, sim.simx_opmode_buffer);
50     if(err==sim.simx_return_ok)
51         fprintf('(roll,pitch,yaw)=(%.1f, %.1f, %.1f)\n',rad2deg(orientation));
52     end
53     [err,signal]=sim.getStringSignal(clientID,'measuredData',sim.simx_opmode_buffer);
54     if (err==sim.simx_return_ok)
55         % Data produced by the child script was retrieved!
56         unpackedsignal = sim.unpackFloats(signal);
57         % disp(unpackedsignal)
58         p = polarscatter(radians,unpackedsignal);
59         p.Marker = ".";
60         drawnow;
61     end
```

Captura de datos del sensor de pose del robot  
(posición y ángulo respecto al eje del mundo)

# Video

CoppeliaSim Edu - scene\_01 - rendering: 1 ms (7.94 fps) - SIMULATION STOPPED

File Edit Add Simulation Tools Modules Scenes Help

Model browser

- modifiers
- sensors
- equipment
- conveyors
- panes
- examples
- block factory
- furniture
- chairs
- plants
- shelves-cupboards
- tables
- household
- infrastructure
- bathroom
- doors
- floors
- other
- urban
- walls
- 20cm high ...
- 240cm high...
- 80cm high ...
- windows
- nature

handrail.ttm

resizable concret block.ttm

Scene hierarchy

- scene\_01 (scene 1)
  - DefaultCamera
  - XYZCameraProxy
  - DefaultLights
  - nakedAckermannSteering
    - rearRightSpringDemp
    - rearLeftSpringDemp
    - frontLeftSpringDemp
    - frontRightSpringDemp
    - lineSegmentRear
    - fastTokuyo
    - robotFrame
  - ResizableFloorMedium
  - ExternalWall
  - worldFrame
  - 20cmHighPillar10cm[0]
  - 20cmHighPillar10cm[1]
  - ConcreteBlock

Bullet 2.78

EDU

Editor - C:\Universidad\ELO325-test\ayudantia\_02\matlab\P2\simpleTest\_v2p1.m

```
sim.simxFinish(-1); % just in case, close all opened connections
clientID=sim.simxStart('127.0.0.1',19999,true,true,5000,5);

if (clientID>-1)
    disp('Connected to remote API server');

% Now retrieve streaming data (i.e. in a non-blocking fashion):
t=clock;
startTime=t(6);
currentTime=t(6);

% Initialize streaming
sim.simxGetInt32Signal(clientID,'keypress', sim.simx_opmode_streaming);
sim.simxGetStringSignal(clientID,'measuredData',sim.simx_opmode_streaming);

[~,refFrame] = sim.simxGetObjectHandle(clientID, '/robotFrame', sim.simx_opmode_blocking);
sim.simxGetObjectPosition(clientID, refFrame, -1, sim.simx_opmode_streaming);
sim.simxGetObjectOrientation(clientID, refFrame, -1, sim.simx_opmode_streaming);

[~,steeringLeft] = sim.simxGetObjectHandle(clientID, '/steeringLeft', sim.simx_opmode_blocking);
[~,steeringRight] = sim.simxGetObjectHandle(clientID, '/steeringRight', sim.simx_opmode_blocking);
[~,motorLeft] = sim.simxGetObjectHandle(clientID, '/motorLeft', sim.simx_opmode_blocking);
[~,motorRight] = sim.simxGetObjectHandle(clientID, '/motorRight', sim.simx_opmode_blocking);
desiredSteeringAngle = 0;
desiredWheelRotSpeed = 0;
steeringAngleDx = 2*pi/180;
wheelRotSpeedDx = 20*pi/180;
d=0.755; %-- 2*d=distance between left and right wheels
l=2.5772; %-- l=distance between front and rear wheels

figure(1)
```

Command Window

New to MATLAB? See resources for [Getting Started](#).

f1 >>

[CoppeliaSim:info] done.

[CoppeliaSim:info] Loading model {C:/Program Files/CoppeliaRobotics/CoppeliaSimEdu/models/infrastructure/other/resizable concret block.ttm}. Serialization version is 24.

[CoppeliaSim:info] File was previously written with CoppeliaSim version 4.05.01 (rev 0).

[CoppeliaSim:info] Model loaded.

[CoppeliaSim:info] Model info:

This model can be modified/adjusted via its customization script.

Input Lua code here, or type "help()" (use ... | Sandbox script

HOME PLOTS APPS EDITOR PUBLISH VIEW FILE NAVIGATE CODE ANALYZE SECTION RUN

Zoom: 100% UIF-8 CRLF simpleTest

Miguel