

Coppelia



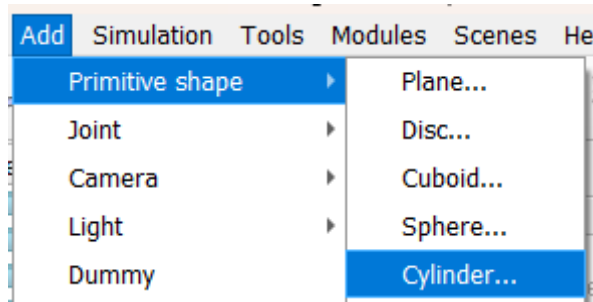
**COPPELIA
ROBOTICS**

Contenido

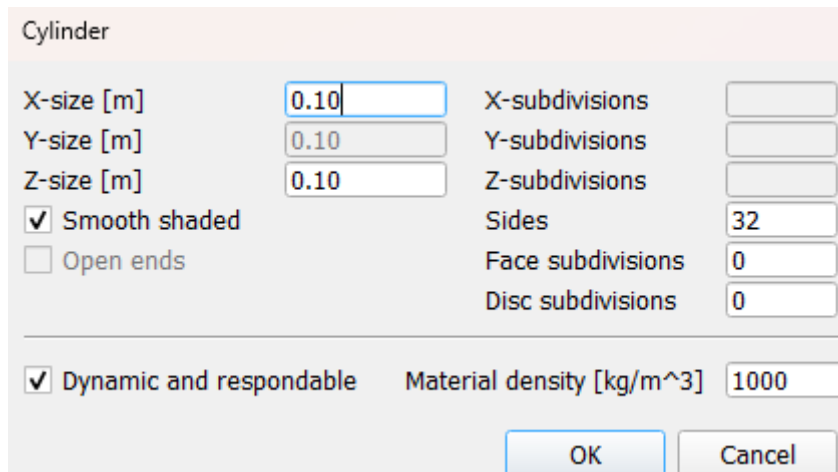
Coppelia.....	1
Diseño de un robot simple	3
Uso de RemoteAPI.....	8

Diseño de un robot simple

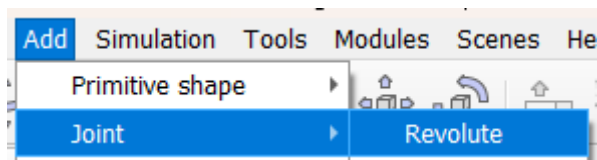
Añadimos un cilindro



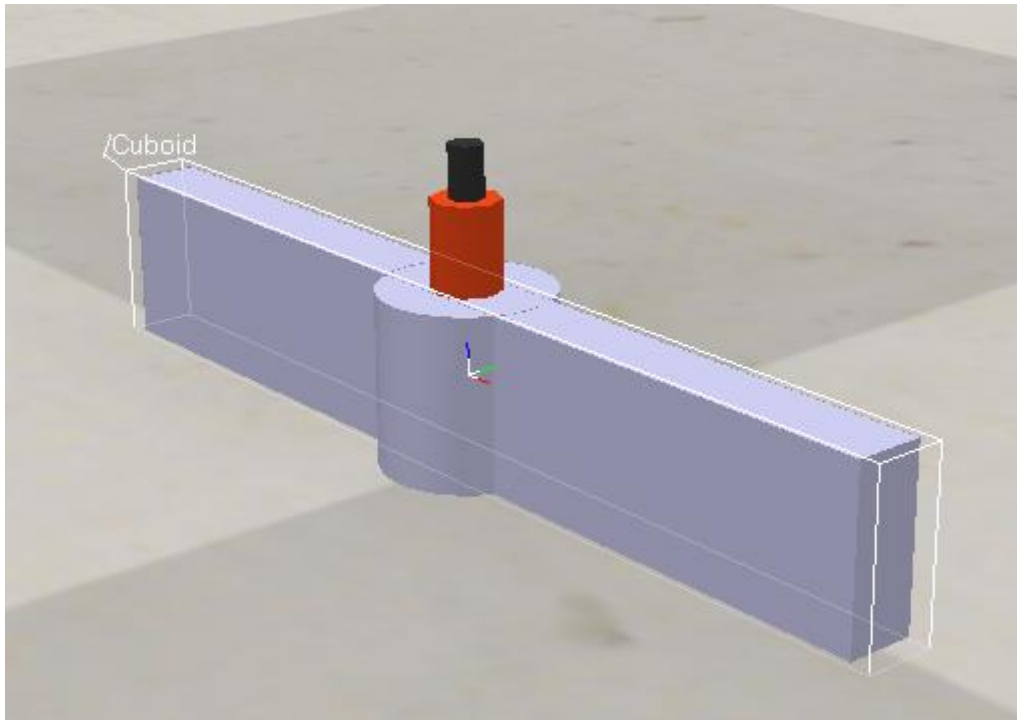
Con las siguientes propiedades:



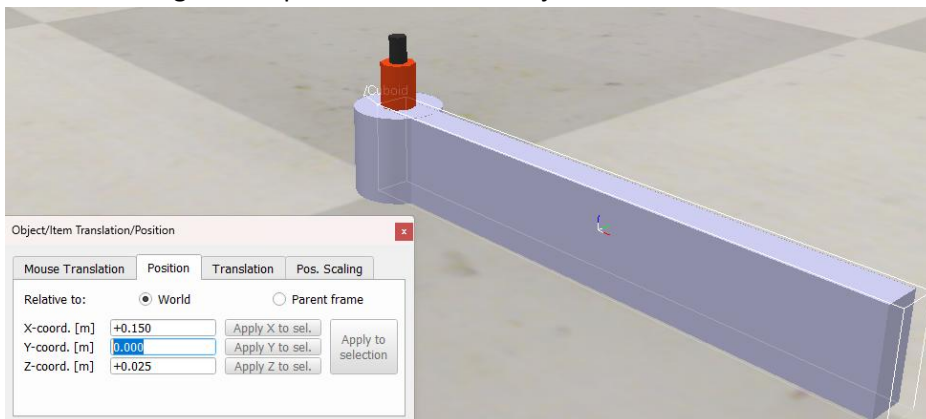
Añadimos una unión de tipo rotacional:



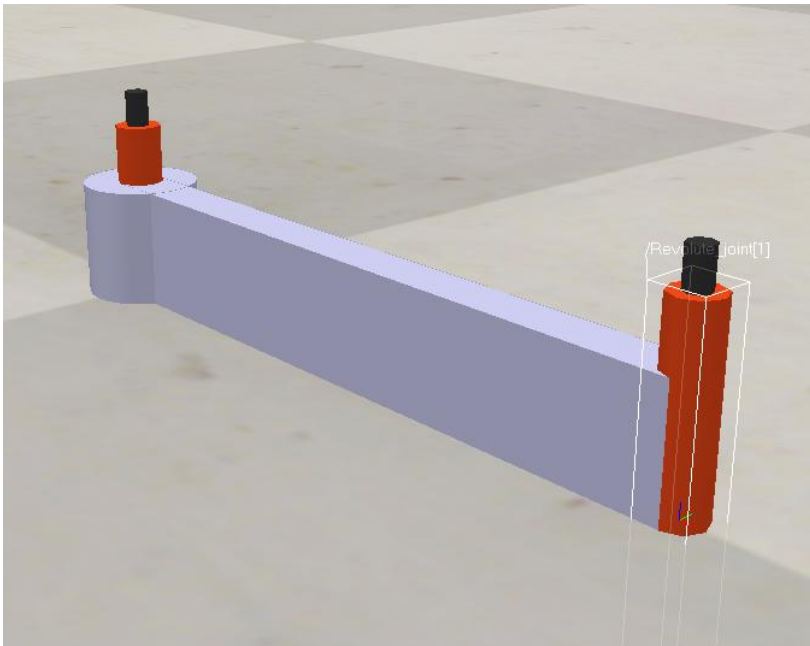
Igual que como hemos hecho hasta ahora, añadimos un cuboide:



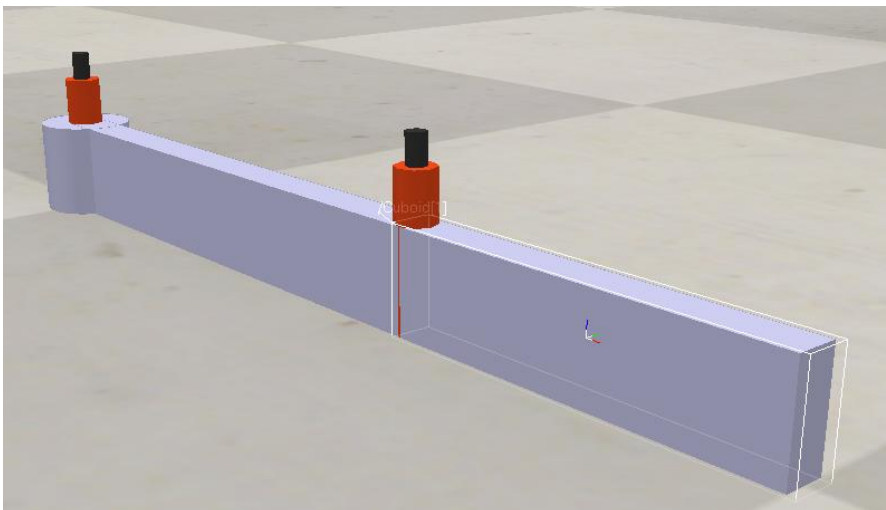
Con esta configuración podemos mover el objeto:



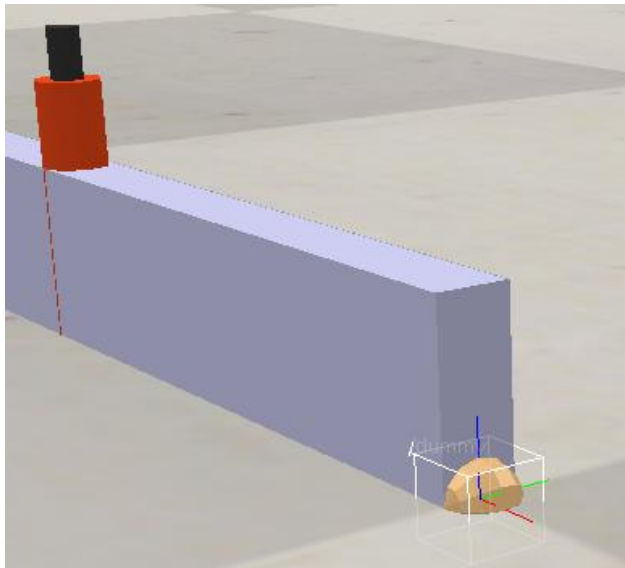
Añadimos otra unión y la ponemos en el otro extremo:



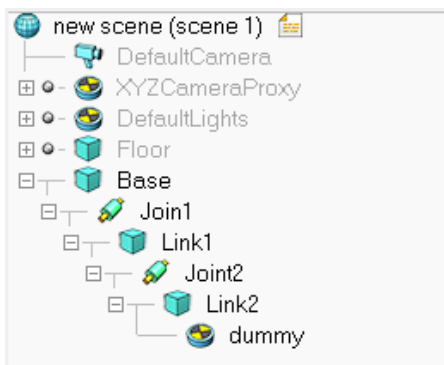
También añadimos otro cuboide, que pondremos en esta última unión:



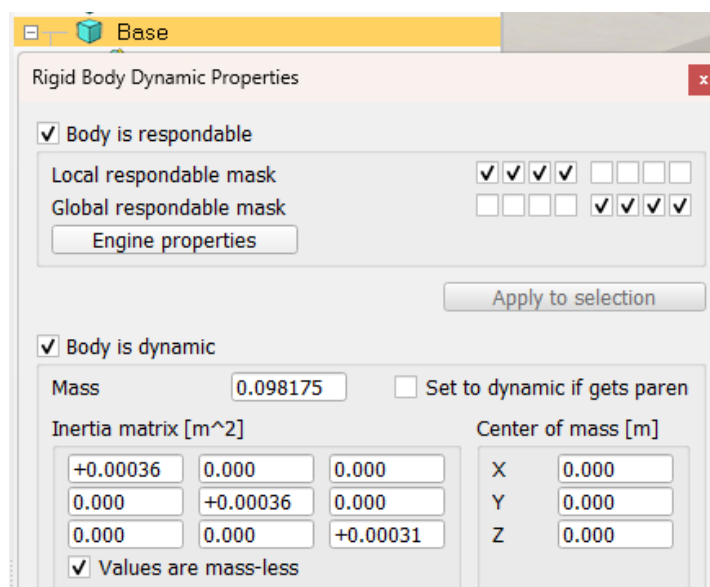
Añadimos un elemento dummy en la punta de este último:



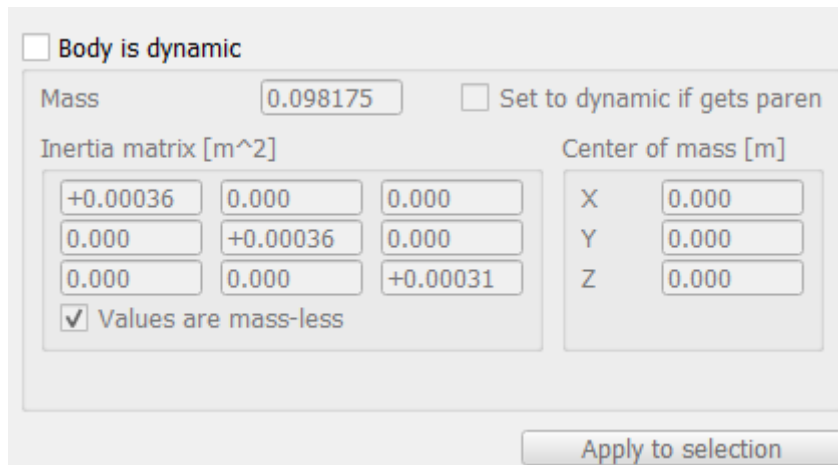
Cambiamos el nombre de nuestras piezas y creamos las correspondientes conexiones para hacer que formen parte del mismo “robot”:



Configuramos las propiedades de la base para que no pueda chocar con los elementos a los que está unida, pero sí con aquellos elementos a los que no y hacemos lo mismo con las otras piezas:



En las propiedades de la base, seleccionamos que no se vea afectada por fuerzas externas, haciendo que quede anclada al suelo:



Uso de RemoteAPI

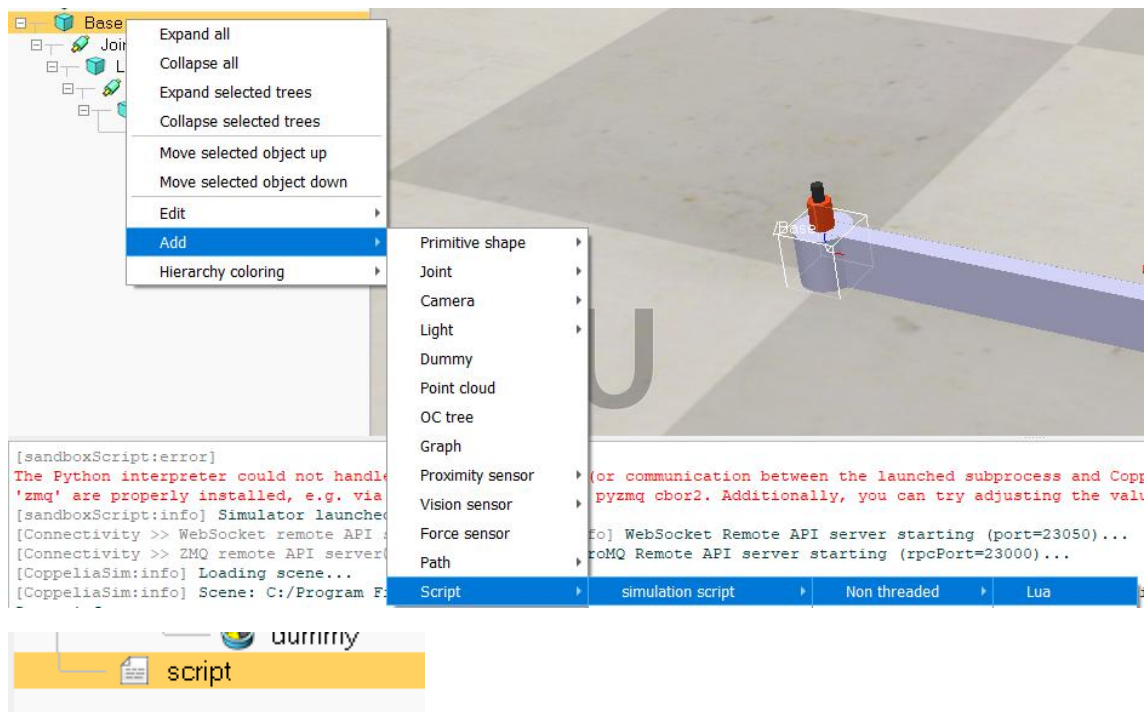
Copiamos en el directorio de nuestro notebook los siguientes archivos:

C:\Program Files\CoppeliaRobotics\CoppeliaSimEdu\programming\legacyRemoteApi\remoteApiBindings\python\python\sim.py

C:\Program Files\CoppeliaRobotics\CoppeliaSimEdu\programming\legacyRemoteApi\remoteApiBindings\python\python\simConst.py

C:\Program Files\CoppeliaRobotics\CoppeliaSimEdu\programming\legacyRemoteApi\remoteApiBindings\lib\lib\Windows\remoteApi.dll

Creamos nuestro script de LUA:




```
Simulation script "/script"
LUA
1 function sysCall_init()
2     sim = require('sim')
3
4     simRemoteApi.start(199999)
5
6 end
7
8 function sysCall_actuation()
9     -- put your actuation code here
10 end
11
12 function sysCall_sensing()
13     -- put your sensing code here
14 end
15
16 function sysCall_cleanup()
17     -- do some clean-up here
18 end
19
20 -- See the user manual or the available code snippets for additional callback functions and de
21
```

Iniciamos la simulación para poder empezar a recibir conexiones:



```
[sandboxScript:info] Simulation started.
```

Y utilizando esta función, podemos conectarnos al puerto que ya hemos definido en el script anterior:

```
def connect(port):
    sim.simxFinish(-1) # just in case, close all opened connections
    clientID=sim.simxStart('127.0.0.1',port,True,True,2000,5) # Conectarse
    if clientID == 0: print("conectado a", port)
    else: print("no se pudo conectar")
    return clientID
```

```
clientID = connect(19999)
[6] ✓ 0.0s
... conectado a 19999
```

Podemos obtener el handler de dummy, los joints y la posición de estos:

```
returnCode,dummy=sim.simxGetObjectHandle(clientID,'Dummy',sim.simx_opmode_blocking)
returnCode,dummyPos=sim.simxGetObjectPosition(clientID, dummy, -1, sim.simx_opmode_blocking)

ret,joint1=sim.simxGetObjectHandle(clientID,'joint1',sim.simx_opmode_blocking)
ret,joint2=sim.simxGetObjectHandle(clientID,'joint2',sim.simx_opmode_blocking)
returnCode,joint1Pos=sim.simxGetJointPosition(clientID, joint1, sim.simx_opmode_blocking)
returnCode,joint2Pos=sim.simxGetJointPosition(clientID, joint2, sim.simx_opmode_blocking)
print(f"dummy: {dummy}, dummy's pos: {dummyPos}\njoint1: {joint1}, joint1 pos: {joint1Pos}\njoint2: {joint2}, joint2 pos: {joint2Pos}")
✓ 0.2s

dummy: 0, dummy's pos: [0.6613926291465759, -0.8682999014854431, 0.6941603422164917]
joint1: 0, joint1 pos: 0.0
joint2: 0, joint2 pos: 0.0
```

También podemos intentar mover las articulaciones, aunque por alguna razón, yo no consigo que se mueva. Probablemente se deba a algún error en la configuración de mi PC.

```
# enviamos la posición de joint1, en radianes.  
q1 = -30 * np.pi/180  
returnCode = sim.simxSetJointTargetPosition(clientID, joint1, q1, sim.simx_opmode_oneshot)  
print(returnCode)  
# enviamos la posición de joint2, en radianes.  
q2 = 30 * np.pi/180  
returnCode = sim.simxSetJointTargetPosition(clientID, joint2, q2, sim.simx_opmode_oneshot)  
print(returnCode)  
✓ 0.0s  
64  
64
```