

GUÍA DE USO COPPELIA SIM -MATLAB 2

OBJETIVO

Crear una aplicación simple que utilice el intercambio de información entre las aplicaciones MATLAB y COPPELIA SIM en ambiente académico.

1. Abrir MATLAB®.
2. Crear el siguiente algoritmo en Matlab y guardarlo en la carpeta de trabajo prog01. Para consultar la referencia de los comandos de Coppelia Sim para MATLAB consultar:

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

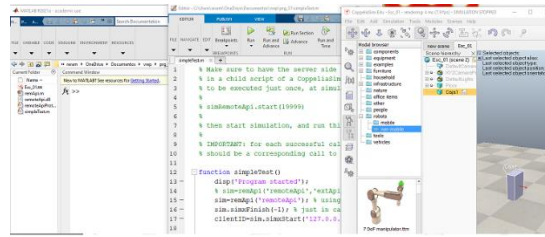
```
%% Programa para verificar el uso de handles y desde MATLAB leer variables en Coppelia Sim.
% Establecer la conexión
vrep=remApi('remoteApi'); % usar el archivo prototipo (remoteApiProto.m)
vrep.simxFinish(-1); % si se requiere, cerrar todas las conexiones abiertas.
% asigna el handle de identificación de cliente clientID
clientID=vrep.simxStart('127.0.0.1',19999,true,true,5000,5);
if (clientID>-1)
    disp('Conexión exitosa')



    %Algoritmo
    % Consulta el handle del objeto Caja1 en la escena Esc01 y lo asigna al handle caja_m.
    [returnCode,caja_m]=vrep.simxGetObjectHandle(clientID,Caja1',vrep.simx_opmode_blocking);
    % Consulta la posición de la caja y la almacena en la variable P
    [returnCode,P]= vrep.simxGetObjectPosition(clientID, caja_m,-1,vrep.simx_opmode_blocking);
    % Muestra en la pantalla de comandos de MATLAB la posición
    disp(P)
    % Consulta la orientación de la caja y la almacena en la variable Or
    [returnCode,Or]= vrep.simxGetObjectOrientation(clientID, caja_m,-
1,vrep.simx_opmode_blocking);
    % Muestra en la pantalla de comandos de MATLAB la orientación
    disp(Or)

    % Termina el programa y cierra la conexión de MATLAB con V-Rep.
    disp('Programa terminado')
    vrep.delete(); % llama el destructor!
```



Guardar el algoritmo.


3. Ordene las ventanas en su pantalla para que pueda ver la ventana de comandos de MATLAB, el script simpleTest.m y la ventana de CoppeliaSim.



4. En CoppeliaSim arrancar la simulación con el ícono: 
5. En Matlab correr el Script.
6. Aceptar si solicita cambio de carpeta.
7. Verificar que en la ventana de MATLAB presenta la posición y la orientación de la Caja1.
8. Parar la simulación en CoppeliaSim:  .

Ejercicios:

1. Usando el ícono  en Coppelia Sim cambie la posición del objeto y con el comando  cambie la orientación, ejecute el ciclo: iniciar la simulación (Coppelia Sim), correr el algoritmo (MATLAB) y parar la simulación (Coppelia Sim). Verifique que en cada caso la información en la ventana de MATLAB corresponde a las propiedades posición y orientación de la Caja1 en la escena de Coppelia Sim.
2. Crear otro objeto en Coppelia Sim, guarde al algoritmo con otro nombre y modifíquelo para que identifique el handle del segundo objeto y luego lea la posición y orientación del segundo objeto y la presente en la ventana de MATLAB.
3. Guarde su escena y ciérrela. Crear una nueva escena y guardarla. Si la guarda en otra carpeta, recordar que hay que crear en esa carpeta copias de los archivos remApi.m, remoteApiProto.m y remoteApi.dll. De las bibliotecas en la parte izquierda superior de Coppelia Sim seleccione robots non-mobile, del conjunto de manipuladores en la parte izquierda inferior busque MTB robot.ttt. Con el mouse arrástrelo y suéltelo en la escena.

Abrir el script con doble click en el ícono  al lado del objeto MTB, abre la ventana Child script. Adicionar debajo de la primera línea function sysCall_init() el siguiente comando simExtRemoteApiStart(19999).

En la ventana Scene Hierarchy expanda el árbol del objeto MTB, encuentra el objeto Rectangle que corresponde a la pieza que es manipulada por MTB. Use el algoritmo de MATLAB ya creado y construya otro algoritmo que lea la velocidad lineal del Rectangle (comando simxGetObjectVelocity) y la muestre en la ventana de MATLAB.