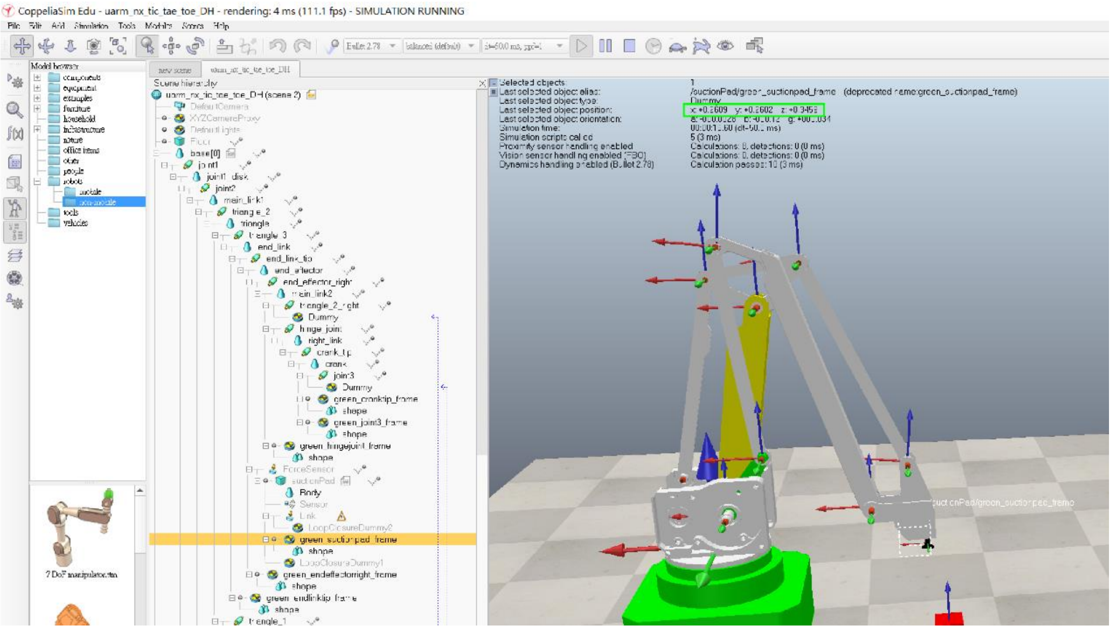
Online Exam

轉動前

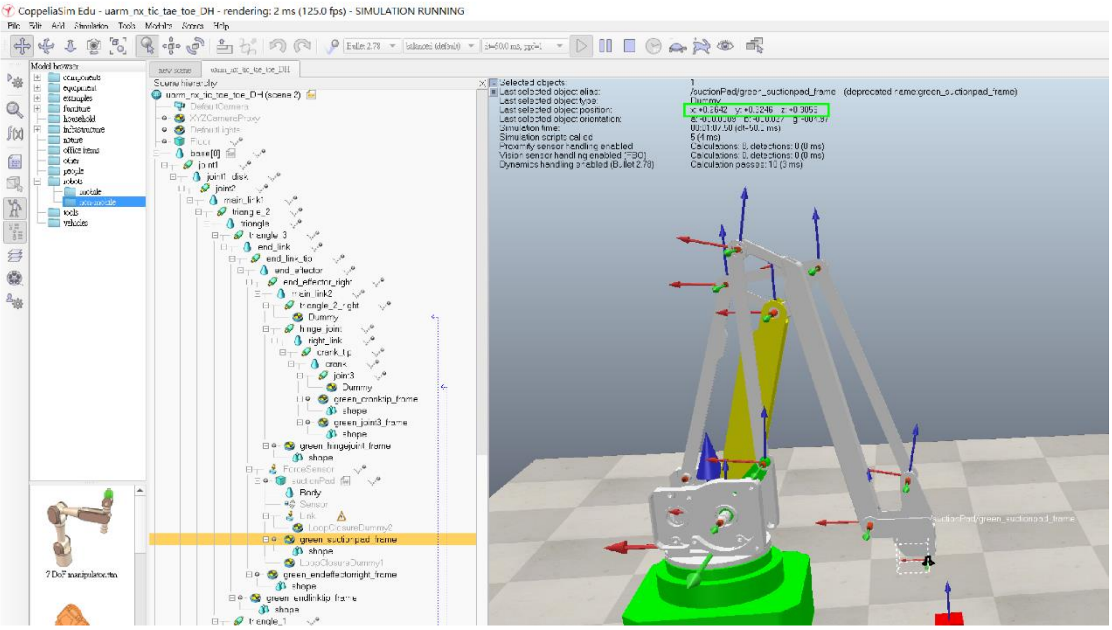


CoppeliaSim Edu - uarm_nx_tic_tae_toe_DH - rendering: 4 ms (111.1 fps) - SIMULATION RUNNING

```
Selected objects:                      1
Last selected object alias:            /suctionPad/green_suctionpad_frame   (deprecated name:green_suctionpad_frame)
Last selected object type:             Dummy
Last selected object position:         x: +0.2609   y: +0.2602   z: +0.3459
Last selected object orientation:      a: -000.0026   b: -000.012   g: +000.034
Simulation time:                       00:00:1[illegible].60 (dt=50.0 ms)
Simulation scripts called              5 (3 ms)
Proximity sensor handling enabled      Calculations: 0, detections: 0 (0 ms)
Vision sensor handling enabled (FBO)   Calculations: 0, detections: 0 (0 ms)
Dynamics handling enabled (Bullet 2.78)  Calculation passes: 10 (3 ms)
```

File Edit Add Simulation Tools Modules Scenes Help
Bullet 2.78
dt=50.0 ms, ppf=1
Model browser
Scene hierarchy
uarm_nx_tic_tae_toe_DH (scene 2)
7 DoF manipulator.ttm
suctionPad/green_suctionpad_frame

各軸轉動 5 度後

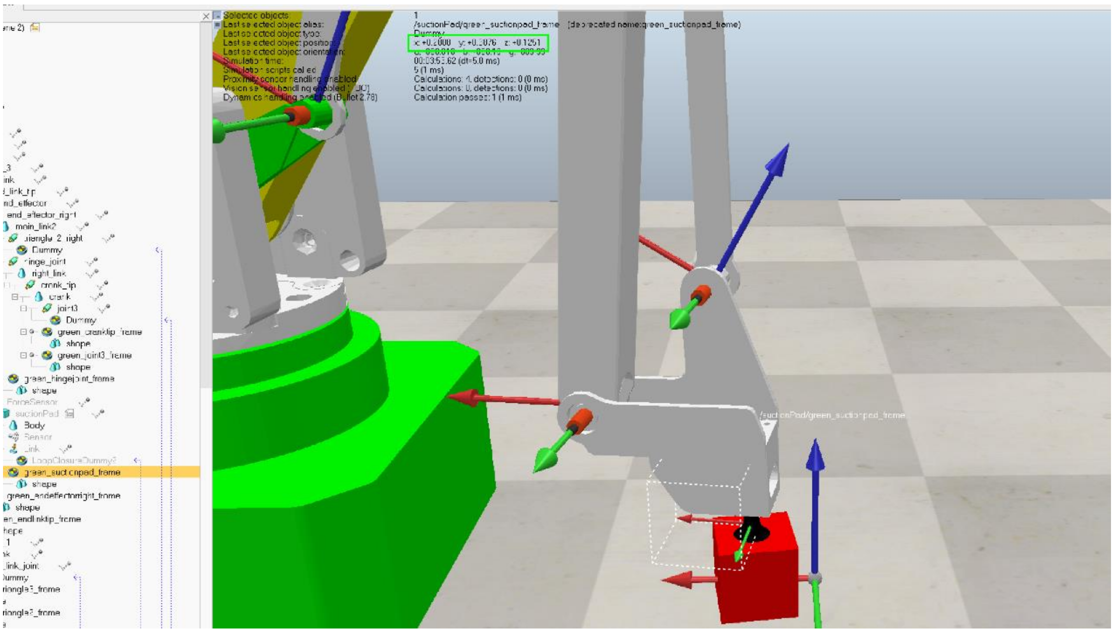


CoppeliaSim Edu - uarm_nx_tic_tae_toe_DH - rendering: 2 ms (125.0 fps) - SIMULATION RUNNING

```
Selected objects:                      1
Last selected object alias:            /suctionPad/green_suctionpad_frame   (deprecated name:green_suctionpad_frame)
Last selected object type:             Dummy
Last selected object position:         x: +0.2642   y: +0.3246   z: +0.3056
Last selected object orientation:      a: -000.0009   b: -000.027   g: -004.97
Simulation time:                       00:01:07.50 (dt=50.0 ms)
Simulation scripts called              5 (4 ms)
Proximity sensor handling enabled      Calculations: 0, detections: 0 (0 ms)
Vision sensor handling enabled (FBO)   Calculations: 0, detections: 0 (0 ms)
Dynamics handling enabled (Bullet 2.78)  Calculation passes: 10 (3 ms)
```

File Edit Add Simulation Tools Modules Scenes Help
Bullet 2.78
dt=50.0 ms, ppf=1
Model browser
Scene hierarchy
uarm_nx_tic_tae_toe_DH (scene 2)
7 DoF manipulator.ttm
suctionPad/green_suctionpad_frame

吸取方塊



```
Selected objects:                          1
Last selected object alias:                /suctionPad/green_suctionpad_frame   (deprecated name:green_suctionpad_frame)
Last selected object type:                 Dummy
Last selected object position:             x: +0.2000   y: +0.2076   z: +0.1251
Last selected object orientation:          a: -090.010   b: -050.10   g: -089.99
Simulation time:                           00:03:53.62 (dt=5.0 ms)
Simulation scripts called:                 5 (1 ms)
Proximity sensor handling enabled:         Calculations: 4, detections: 0 (0 ms)
Vision sensor handling enabled (FBO):      Calculations: 0, detections: 0 (0 ms)
Dynamics handling enabled (Bullet 2.78):   Calculation passes: 1 (1 ms)
```

```lua
function sysCall_init()
    axis1=sim.getObject('./joint1')
    axis2=sim.getObject('./joint2')
    axis3=sim.getObject('./joint3')
    suctionPad=sim.getObject('/suctionPad')
    rotation1 = 0
    rotation2 = 0
    rotation3 = 0
    deg = math.pi/180.
    enableSuctionPad(true)
end
function enableSuctionPad(enable)
    -- use the suctionPad object to pass the variable activity value
    -- if enable = true, the activity = 'on'
    if enable then
        sim.writeCustomDataBlock(suctionPad,'activity','on')
    else
        sim.writeCustomDataBlock(suctionPad,'activity','off')
    end
end
function sysCall_actuation()
--[[
    rotation1 = rotation1 + 1*deg
    print(rotation1)
    sim.setJointPosition(axis1, -rotation1)
    sim.setJointPosition(axis2, -rotation1)
]]
    message, auxiliaryData=sim.getSimulatorMessage()
    while message ~= -1 do
        key=auxiliaryData[1]
        sim.addStatusbarMessage('user press key:'..key)
        if (message==sim.message_keypress) then
           --if (auxiliaryData[1]==112) then --p activate the suction pad
            if (auxiliaryData[1]==string.byte('p')) then
                -- if key p pressed activate the suction mode
--sim.setScriptSimulationParameter(sim.getScriptAssociatedWithObject(suctionPad),'active','true')
                enableSuctionPad(true)
            end -- if p
            if (auxiliaryData[1]==string.byte('q')) then --q deactivate the suction pad
                -- if key q pressed deactivate the suction mode
--sim.setScriptSimulationParameter(sim.getScriptAssociatedWithObject(suctionPad),'active','false')
                enableSuctionPad(false)
            end -- if q
            if (auxiliaryData[1]==string.byte('k')) then --k right turn in degree
                -- if key k pressed axis1 angle adds 2 degrees
                 rotation1 = rotation1 + 2*deg
                 --sim.setJointPosition(axis1, rotation1)
                 sim.setJointTargetPosition(axis1, rotation1)
            end -- if k
            if (auxiliaryData[1]==string.byte('l')) then --l left turn in degree
                -- if key l pressed axis1 angle substract 2 degrees
                 rotation1 = rotation1 - 2*deg
                 --sim.setJointPosition(axis1, rotation1)
                 sim.setJointTargetPosition(axis1, rotation1)
            end -- if l
-- for joint2 start ##########################################################
            if (auxiliaryData[1]== string.byte("h")) then --r right turn in degree
                -- if key r pressed axis1 angle adds 2 degrees
                 rotation2 = rotation2 + 1*deg
                 --sim.setJointPosition(axis2, rotation2)
                 sim.setJointTargetPosition(axis2, rotation2)
            end -- if k
            if (auxiliaryData[1]==string.byte("j")) then
                -- if key j pressed axis2 angle substract 2 degrees
                 rotation2 = rotation2 - 1*deg
                 --sim.setJointPosition(axis2, rotation2)
                 sim.setJointTargetPosition(axis2, rotation2)
            end -- if j
-- for joint2 end ##########################################################
-- for joint3 start ##########################################################
            if (auxiliaryData[1]== string.byte("f")) then
                -- if key f pressed axis3 angle adds 2 degrees
                 rotation3 = rotation3 + 1*deg
                 --sim.setJointPosition(axis3, rotation3)
                 sim.setJointTargetPosition(axis3, rotation3)
            end -- if f
            if (auxiliaryData[1]==string.byte("g")) then
                -- if key g pressed axis3 angle substract 2 degrees
                 rotation3 = rotation3 - 1*deg
                 --sim.setJointPosition(axis3, rotation3)
                 sim.setJointTargetPosition(axis3, rotation3)
            end -- if g
-- for joint3 end ##########################################################
            if (auxiliaryData[1]==99) then --c coordinate of block
                blockPosition = sim.getObjectPosition(block, BaseFrame)
                sim.addStatusbarMessage("coordinate:"..table_to_string(blockPosition))
            end --if c
            if (auxiliaryData[1]==string.byte('z')) then
                 rotation1 = rotation1 - 5*deg
                 rotation2 = rotation2 - 5*deg
                 rotation3 = rotation3 - 5*deg
                 sim.setJointTargetPosition(axis1, rotation1)
                 sim.setJointTargetPosition(axis2, rotation2)
                 sim.setJointTargetPosition(axis3, rotation3)
            end -- if z
             if (auxiliaryData[1]==string.byte('x')) then
                 rotation1 = rotation1 - 10*deg
                 rotation2 = rotation2 - 27*deg
                 rotation3 = rotation3 - 29*deg
                 sim.setJointTargetPosition(axis1, rotation1)
                 sim.setJointTargetPosition(axis2, rotation2)
                 sim.setJointTargetPosition(axis3, rotation3)
            end -- if x
       end  -- if
    message, auxiliaryData=sim.getSimulatorMessage()
    end -- while
end -- function

function sysCall_sensing()
--[[
    -- Read Proximity sensor (0= nothing detected, 1 = object detected)
    local res = sim.readProximitySensor(proximity)

    -- Check if possible to insert an new box
    if (sim.getSimulationTime()-T_last_inserted > T_insert) and not hasStopped then
        insertBox()
    end

    -- If proximity sensor detects an object, stop the belt, stop inserting objects
    if res == 1 and not hasStopped then
        if boolList[1] then
            sim.setScriptSimulationParameter(sim.handle_self,"conveyorBeltVelocity",0)
            deltaTime = sim.getSimulationTime()-T_last_inserted
            hasStopped = true
        else
            local box = table.remove(boxList,1)
            local boxDummy = table.remove(boxDummyList,1)
            table.remove(boolList,1)

            sim.removeObject(box)
            sim.removeObject(boxDummy)
        end
    end

    -- If proximity sensor detects nothing and belt has stopped, start belt, continue inserting
    if res == 0 and hasStopped then
        sim.setScriptSimulationParameter(sim.handle_self,"conveyorBeltVelocity",beltSpeed)
        hasStopped = false
```

```lua
        T_last_inserted = sim.getSimulationTime()-deltaTime
    end

]]--
end
function sysCall_sensing()
    -- put your sensing code here
end
function sysCall_cleanup()
    -- do some clean-up here
end
-- Convert a lua table into a lua syntactically correct string
function table_to_string(tbl)
    local result = "{"
    for k, v in pairs(tbl) do
        -- Check the key type (ignore any numerical keys - assume its an array)
        if type(k) == "string" then
            result = result.."[\""..k.."\"]".."="
        end
        -- Check the value type
        if type(v) == "table" then
            result = result..table_to_string(v)
        elseif type(v) == "boolean" then
            result = result..tostring(v)
        else
            v = round(v, 4)
            result = result.."\""..v.."\""
        end
        result = result..","
    end
    -- Remove leading commas from the result
    if result ~= "" then
        result = result:sub(1, result:len()-1)
    end
    return result.."}"
end
function round(x, n)
    n = math.pow(10, n or 0)
    x = x * n
    if x >= 0 then x = math.floor(x + 0.5) else x = math.ceil(x - 0.5) end
    return x / n
end
function insertBox()
    -- Generate random numbers
    local rand1 = math.random()
    local rand2 = math.random()
    local rand3 = math.random()
    -- Generate random disturbances on position and orientation
    local dx = (2*rand1-1)*0.1
    local dy = (2*rand2-1)*0.1
    local dphi = (2*rand3-1)*0.5
    local disturbedCoordinates = {0,0,0}
    disturbedCoordinates[1] = insertCoordinate[1]+dx
    disturbedCoordinates[2] = insertCoordinate[2]+dy
    disturbedCoordinates[3] = insertCoordinate[3]
    -- Copy and paste box and boxDummy
    local insertedObjects = sim.copyPasteObjects({box,boxDummy},0)
    -- Update last inserted box time
    T_last_inserted = sim.getSimulationTime()
    -- Move and rotate
    sim.setObjectPosition(insertedObjects[1],-1,disturbedCoordinates)
    sim.setObjectOrientation(insertedObjects[1],-1,{0,0,dphi})
    -- Store handles to boxes and dummies
    table.insert(boxList,insertedObjects[1])
    table.insert(boxDummyList,insertedObjects[2])
    -- Decide if object is good or bad
    local decision = math.random()
    if decision <= goodPercentage then
    -- Object is good, assign goodColor
        sim.setShapeColor(insertedObjects[1],nil,sim.colorcomponent_ambient_diffuse,goodColor)
        table.insert(boolList,true)
    else
    -- Object is bad, assign random color
        sim.setShapeColor(insertedObjects[1],nil,sim.colorcomponent_ambient_diffuse,{rand1,rand2,rand3})
        table.insert(boolList,false)
    end
```

```
end
-- See the user manual or the available code snippets for additional callback functions and details
```