## Basic:

* Unable to figure out how to use the vision sensor? Why not try double clicking on the vision sensor icon in the scene hierarchy and experimenting around with the properties?
* Unable to figure out where to write the code? Try learning about child scripts and what they do.
* Unable to figure out how to write the code? Simply scroll through the various regular API functions in CoppeliaSim. You might encounter some interesting ones.

## Explicit:

* Help with vision sensor:
  + <https://www.coppeliarobotics.com/helpFiles/en/visionSensors.htm>
  + <https://www.coppeliarobotics.com/helpFiles/en/visionSensorPropertiesDialog.htm>
  + <https://www.coppeliarobotics.com/helpFiles/en/visionSensorDescription.htm>
* Help with child script:
  + <https://www.coppeliarobotics.com/helpFiles/en/childScripts.htm>
  + <https://coppeliarobotics.com/helpFiles/en/simulationScripts.htm>
* Help with the code:
  + <https://www.coppeliarobotics.com/helpFiles/>
  + <https://www.coppeliarobotics.com/helpFiles/en/regularApi/simHandleVisionSensor.htm>
  + <https://www.coppeliarobotics.com/helpFiles/en/accessingSceneObjects.htm>
  + <https://www.coppeliarobotics.com/helpFiles/en/regularApi/simSetStringSignal.htm>

## Walkthrough:

* First, open Vision\_sensor’s scene object properties by double-clicking on its icon in the scene hierarchy.
* Then set ‘Near/ far clipping plane’ to 1.00e-02 / 1.50e+00, ‘ortho. size’ to 00.1000.
* Right-click on ‘Vision\_sensor’ in scene hierarchy🡺Add🡺Associated child script🡺Non-threaded🡺Lua
* Double click on the script icon to open it.
* Copy-paste this code into it-

function sysCall\_init()

-- do some initialization here

vs=sim.getObject(".")

end

function sysCall\_actuation()

-- put your actuation code here

end

function sysCall\_sensing()

-- put your sensing code here

detectionCount, auxiliaryValuePacket1= sim.handleVisionSensor(vs)

if ((auxiliaryValuePacket1[12]>=0.6 and auxiliaryValuePacket1[12]<=0.7) and (auxiliaryValuePacket1[13]>=0.35 and auxiliaryValuePacket1[13]<=0.45)and (auxiliaryValuePacket1[14]>=0 and auxiliaryValuePacket1[14]<=0.05)) then

--print('orange')

sim.setStringSignal('colorDetected','orange')

--print(auxiliaryValuePacket1[12],auxiliaryValuePacket1[13],auxiliaryValuePacket1[14])

elseif ((auxiliaryValuePacket1[12]>=0.0 and auxiliaryValuePacket1[12]<=0.05) and (auxiliaryValuePacket1[13]>=0.6 and auxiliaryValuePacket1[13]<=0.7)and (auxiliaryValuePacket1[14]>=0 and auxiliaryValuePacket1[14]<=0.05)) then

--print('green')

sim.setStringSignal('colorDetected','green')

--print(auxiliaryValuePacket1[12],auxiliaryValuePacket1[13],auxiliaryValuePacket1[14])

else

sim.setStringSignal('colorDetected','unknown')

--print(auxiliaryValuePacket1[12],auxiliaryValuePacket1[13],auxiliaryValuePacket1[14])

end

end

function sysCall\_cleanup()

-- do some clean-up here

end

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