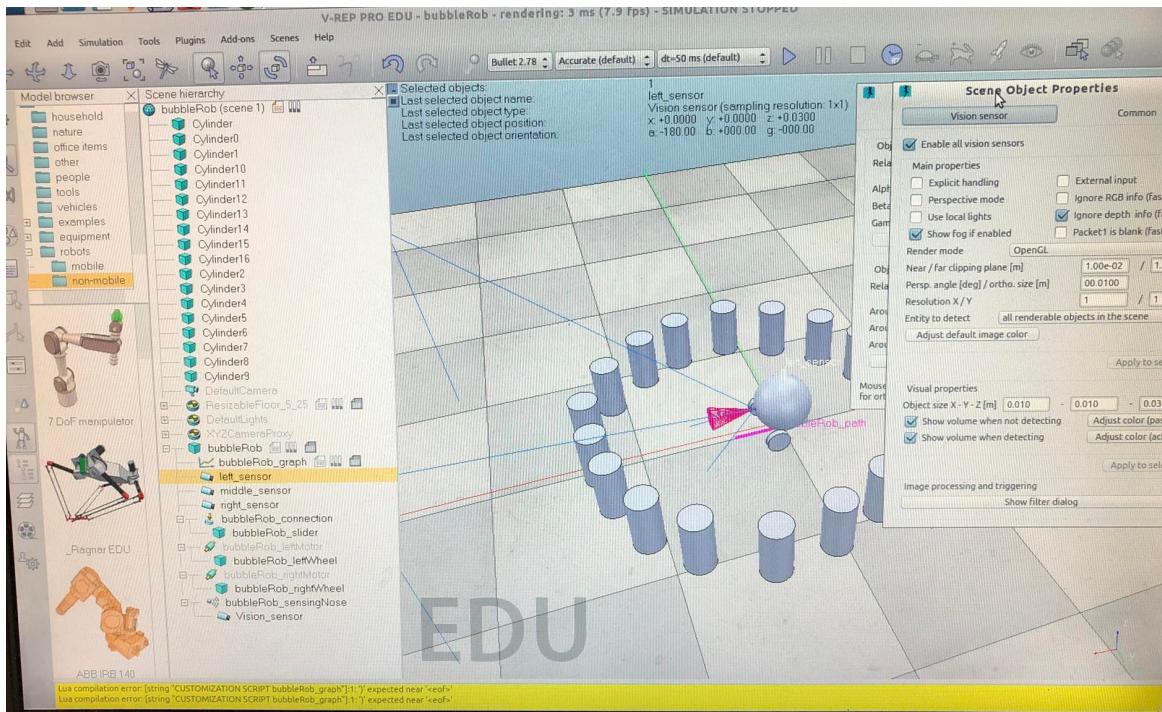


Sprawozdanie

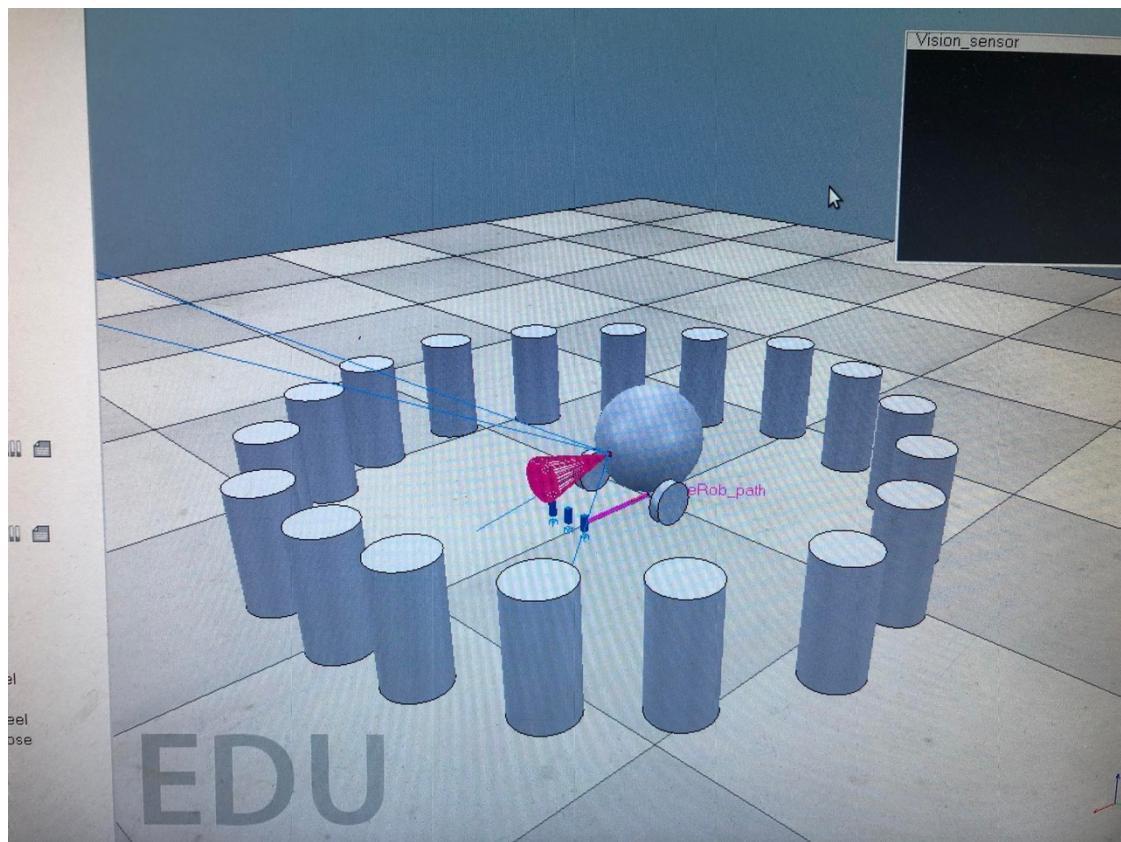
Kamil Warchoł

Celem zajęć było zbudowanie na podstawie robota z pierwszego laboratorium LineFollowera.

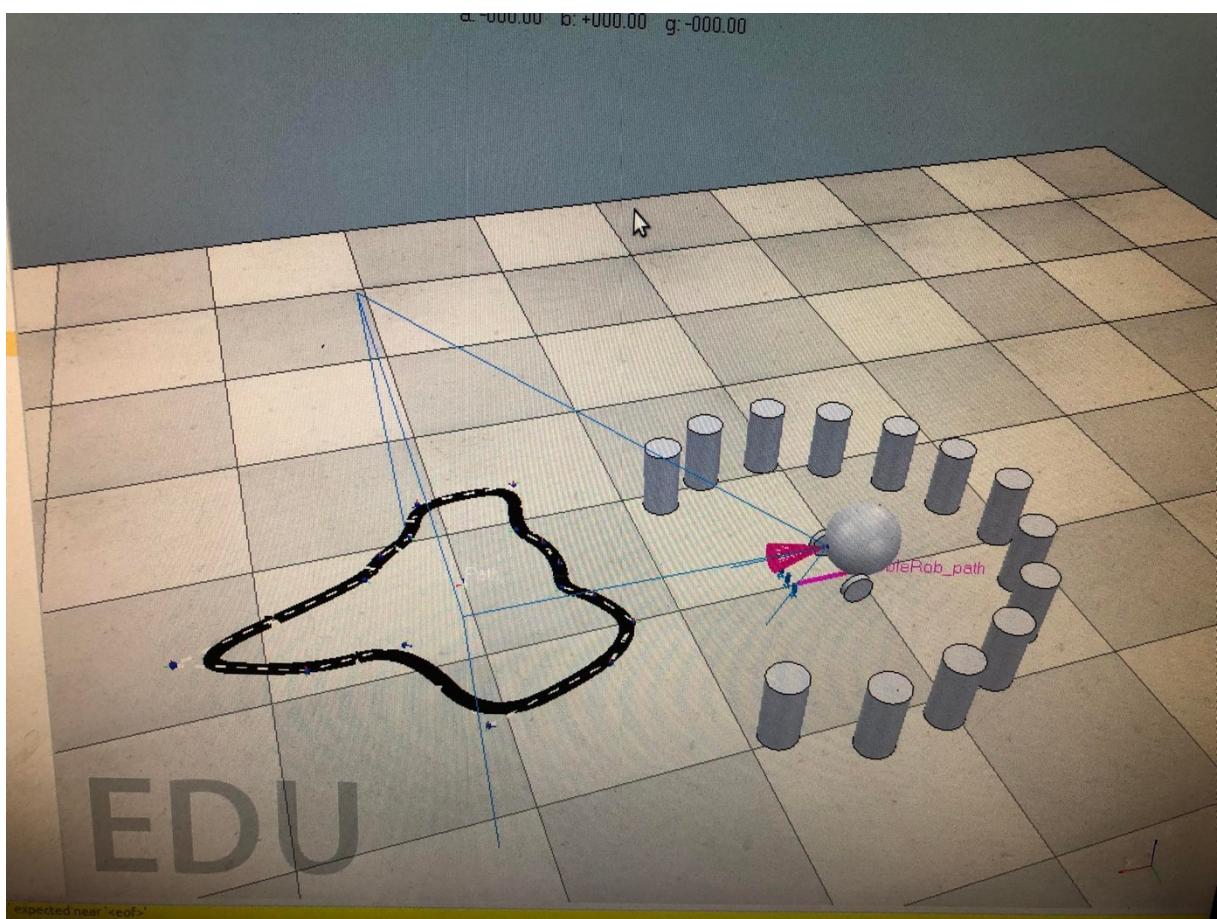
Zaczynamy od załadowania sceny przygotowanej na pierwszym laboratorium i dodaniu czujników wizyjnych:



Kolejnym etapem jest odpowiednia konfiguracja czujników oraz ich ustawienie w odpowiednich pozycjach:



Później usuwamy zbędne cylindry, aby BubbleRob mógł wyjechać z otaczającego go koła oraz tworzymy i konfigurujemy ścieżkę, którą nasz LineFollower ma śledzić:



Ostatnim etapem jest zmodyfikowanie skryptu w którym zdefiniujemy działanie LineFollowera:

Skrypt:

```
function speedChange_callback(ui,id,newVal)
    speed=minMaxSpeed[1]+(minMaxSpeed[2]-minMaxSpeed[1])*newVal/100
end
if (sim_call_type==sim_childscriptcall_initialization) then
    -- This is executed exactly once, the first time this script is executed
    bubbleRobBase=simGetObjectAssociatedWithScript(sim_handle_self) --this is
    bubbleRob's handle
    leftMotor=simGetObjectHandle("bubbleRob_leftMotor") -- Handle of the left
    motor
    rightMotor=simGetObjectHandle("bubbleRob_rightMotor") -- Handle of the
    right motor
    noseSensor=simGetObjectHandle("bubbleRob_sensingNose") -- Handle of the
    proximity sensor
    minMaxSpeed={50*math.pi/180,300*math.pi/180} -- Min and max speeds for
    each motor
```

```

backUntilTime=-1 -- Tells whether bubbleRob is in forward or backward mode

floorSensorHandles={-1,-1,-1}
floorSensorHandles[1] = simGetObjectHandle("left_sensor")
floorSensorHandles[2] = simGetObjectHandle("middle_sensor")
floorSensorHandles[3] = simGetObjectHandle("right_sensor")
-- Create the custom UI:
xml = '<ui title="'.simGetObjectName(bubbleRobBase)..''
speed"closeable="false" resizeable="false" activate="false">..[[[
<hslider minimum="0" maximum="100" onchange="speedChange_callback" id="1"/>

<label text="" style="* {margin-left: 300px; }"/>
</ui>
]]
ui=simExtCustomUI_create(xml)
speed=(minMaxSpeed[1]+minMaxSpeed[2])*0.5
simExtCustomUI_setSliderValue(ui,1,100*(speed-minMaxSpeed[1])/(
minMaxSpeed[2]-minMaxSpeed[1]))
end
if (sim_call_type==sim_childscriptcall_actuation) then
linesensor = {false,false,false}
for i=1,3,1 do
    result,data=simReadVisionSensor(floorSensorHandles[i])
    if (result >= 0) then
        linesensor[i] = (data[11]<0.33)
    end
    print(linesensor[i])
end
rightV = speed
leftV = speed
if (linesensor[3]) then
    rightV=0.003*speed
end
if (linesensor[1]) then
    leftV=0.003*speed
end
if (linesensor[1] and linesensor[3]) then
    backUntilTime = simGetSimulationTime()+2
end
result=simReadProximitySensor(noseSensor) -- Read the proximity sensor
-- If we detected something, we set the backward mode:
if (result>0) then backUntilTime=simGetSimulationTime()+4 end
if (backUntilTime<simGetSimulationTime()) then
-- When in forward mode, we simply move forward at the desired speed
simSetJointTargetVelocity(leftMotor,leftV)
simSetJointTargetVelocity(rightMotor,rightV)
else
-- When in backward mode, we simply backup in a curve at reduced speed
simSetJointTargetVelocity(leftMotor,-speed/2)
simSetJointTargetVelocity(rightMotor,-speed/8)
end
end
if (sim_call_type==sim_childscriptcall_cleanup) then
    simExtCustomUI_destroy(ui)
end

```