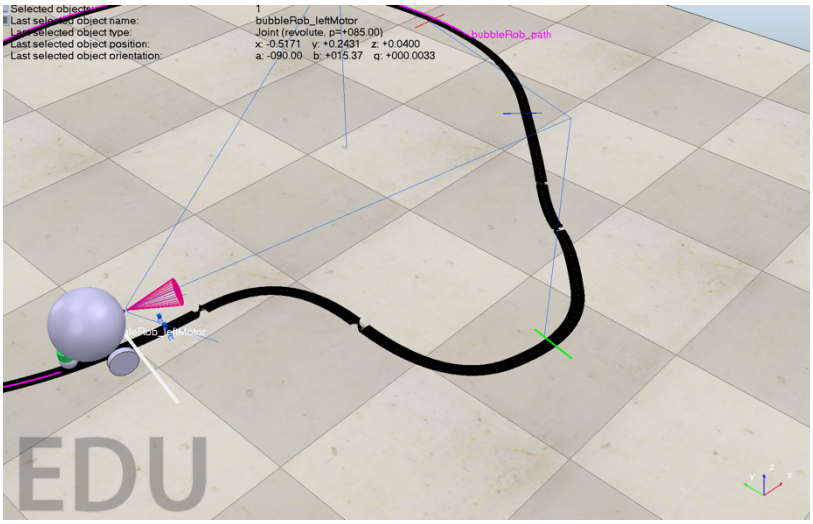
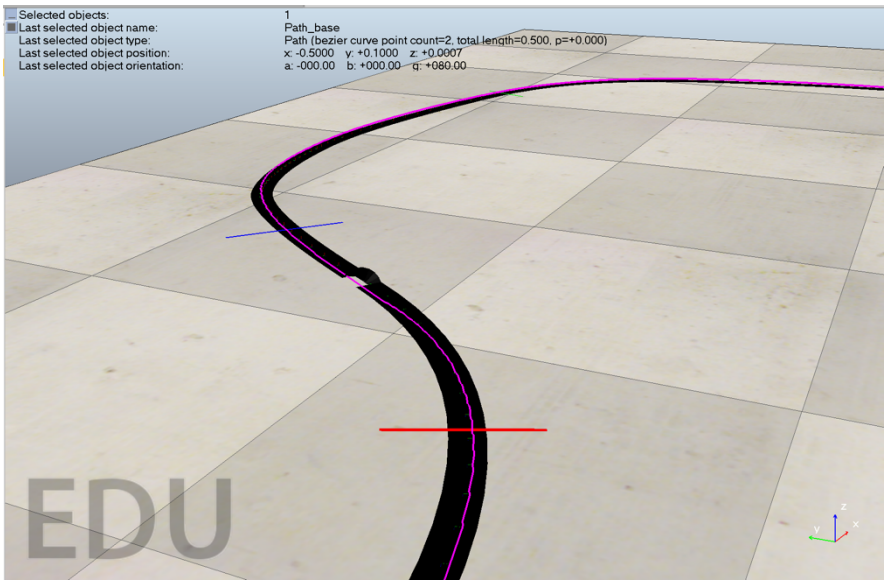
Sprawozdanie
Kamil Warchoł

Celem zajęć było zbudowanie prototypu systemu magazynowego na podstawie LineFollowera wykonanego na ostatnich zajęciach.
Mój pomysł opierał się na magazynie zbudowanym na planie zamkniętej ścieżki na której porusza się robot (lub kilka robotów). Stacje odbioru mają różne kolory, przez co możemy dokładnie ustalić na której stacji zatrzyma się robot, zadając mu zatrzymanie po zidentyfikowaniu danej barwy. Po wykryciu koloru robot zatrzymuje się na 5 sekund i jedzie dalej, dopóki nie ujrzy ponownie zadanej barwy lub barwy bazy (magazynu).



Linefollower wraz z fragmentem trasy – widoczna basa w postaci białej linii oraz kolejne stacje o różnych barwach.



Zbliżenie na fragment trasy wraz z widocznymi różnokolorowymi stacjami.

Skrypt:

```
function speedChange_callback(ui,id,newVal)
 speed=minMaxSpeed[1]+(minMaxSpeed[2]-minMaxSpeed[1])*newVal/100
end

function wait(seconds)
   local start=os.time()
   repeat until os.time()>start+seconds
end

if (sim_call_type==sim.syscb_init) then
 -- This is executed exactly once, the first time this script is executed
 bubbleRobBase=sim.getObjectAssociatedWithScript(sim.handle_self) --this is bubbleRob's handle
 leftMotor=sim.getObjectHandle("bubbleRob_leftMotor") -- Handle of the left motor
 rightMotor=sim.getObjectHandle("bubbleRob_rightMotor") -- Handle of the right motor
 noseSensor=sim.getObjectHandle("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]=sim.getObjectHandle("leftSensor")
   floorSensorHandles[2]=sim.getObjectHandle("MiddleSensor")
   floorSensorHandles[3]=sim.getObjectHandle("rightSensor")

-- Create the custom UI:
 xml = '<ui title="'..sim.getObjectName(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=simUI.create(xml)
 speed=(minMaxSpeed[1]+minMaxSpeed[2])*0.5
 simUI.setSliderValue(ui,1,100*(speed-minMaxSpeed[1])/
(minMaxSpeed[2]-minMaxSpeed[1]))
end
if (sim_call_type==sim.syscb_actuation) then

 result=sim.readProximitySensor(noseSensor) -- Read the proximity sensor
 -- If we detected something, we set the backward mode:
 if (result>0) then backUntilTime=sim.getSimulationTime()+4 end

   linesensor={false,false,false}
   greensensor ={false,false,false}
   bluesensor ={false,false,false}
   redsensor ={false,false,false}
   basesensor ={false,false,false}
   color = 2 --1 dla niebieskiego, 2 zielony, 3 czerowony

      for i=1,3,1 do
      result,data=sim.readVisionSensor(floorSensorHandles[i])
         if (result>=0) then
            linesensor[i]=(data[11]<0.33)
            greensensor[i]=(data[13]>0.95)
```

```
            bluesensor[i]=(data[14]>0.95)
            redsensor[i]=(data[12]>0.95)
            basesensor[i]=(data[11]>0.95)
          end
      -- print(data[13])
       end

   rightV=speed
   leftV=speed

   if linesensor[1] then
      leftV=0.005*speed
   end
   if linesensor[3] then
      rightV=0.005*speed
   end
   if linesensor[1] and linesensor[3] then
      backUntilTime=sim.getSimulationTime()+2
   end
   if greensensor[2] and color == 2 then
     wait(5)
   end
   if bluesensor[2] and color == 1 then
     wait(5)
   end
   if redsensor[2] and color == 3 then
     wait(5)
   end
   if basesensor[2] then
      leftV=0
      rightV=0
   end



 if (backUntilTime<sim.getSimulationTime()) then
 -- When in forward mode, we simply move forward at the desired speed
 sim.setJointTargetVelocity(leftMotor,leftV)
 sim.setJointTargetVelocity(rightMotor,rightV)
 else
 -- When in backward mode, we simply backup in a curve at reduced speed
 sim.setJointTargetVelocity(leftMotor,-speed/2)
 sim.setJointTargetVelocity(rightMotor,-speed/8)
 end
end
if (sim_call_type==sim.syscb_cleanup) then
 simUI.destroy(ui)
end
```