

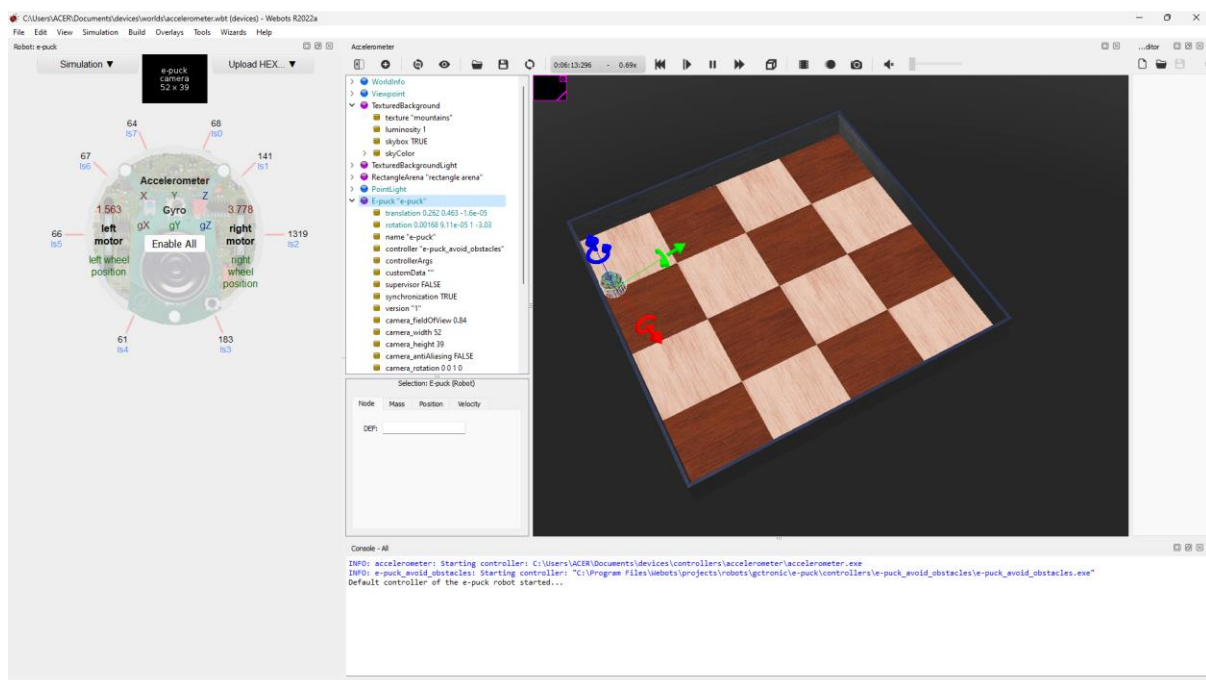
Nama : Nurul Amelia

NIM : 1103194032

### [Tugas – 1]

Robot e-puck.

Pada robot e-puck mempunyai 2 motor yang berada disebalah kanan dan kiri (right motor and left motor) yang berfungsi sebagai *actuator* pada robot tersebut. Motor sebagai *actuator* untuk memuat roda robot. Selain itu, robot e-puck juga mempunyai beberapa sensor diantaranya adalah sensor *accelerometer*, yang berfungsi sebagai sensor untuk mengukur percepatan atau perubahan kecepatan dari robotnya. Kemudian, terdapat beberapa sensor seperti Ultrasound Sensor yang digunakan untuk mengukur jarak (terdapat 8 buah sensor ultrasound yang terdapat pada robot e-puck). Pada robot e-puck juga terdapat kamera dengan resolusi 52 x 39 pixel.



Selection: E-puck (Robot)			
Node	Mass	Position	Velocity
excluding descendants			
mass:	0.15 kg		
density:	702.452 kg/m^3		
CoM:	relative	0 0 0.015	
		9.78585e-05 0 0	
Inertia matrix:		0 8.64333e-05 0	
		0 0 8.7486	

Robot e-puck mempunyai berat atau massa sebesar 0,15 kg.

Langkah-langkah menjalankan robot e-puck, sebagai berikut :

1. Membuat sebuah *controller*.
2. Mencoba mengirimkan sinyal ke robot e-puck lewat right motor dan left motor.

Deklarasi :

Pertama untuk mengetahui nama *actuator*, klik kanan pada e-puck dan pilih View Generated PROTO Node. Source code yang ditampilkan seperti gambar dibawah ini :

```
Robot {
  translation IS translation
  rotation IS rotation
  children [
    HingeJoint {
      jointParameters HingeJointParameters {
        axis 0 1 0
        anchor 0 0 0.02
      }
      device [
        RotationalMotor {
          name "left wheel motor"
          consumptionFactor -0.001 # small trick to encourage the movement (calibrated

          maxVelocity 6.28 # 1000 step/s

        }
        PositionSensor {
          name "left wheel sensor"
          resolution 0.00628 # (2 * pi) / 1000
        }
      ]
    }
  ]
  endPoint DEF EPUCK_LEFT_WHEEL Solid {
    translation 0 0.026 0.02
    children [
      DEF EPUCK_WHEEL Transform {
        rotation 1 0 0 -1.570796
        children [
          Shape {
            appearance DEF EPUCK_TRANSPARENT_APPEARANCE PBRAppearance {
              baseColor 0.5 0.5 0.5
              transparency 0.4
              roughness 0.5
              metalness 0
            }
            geometry Cylinder {
              height 0.003
              radius 0.02
              subdivision 24
            }
          }
          Transform {
            translation 0 0 0.0016
            rotation 1 0 0 1.570796
            children [
```

Hasil deklarasi :

```
WbDeviceTag right_motor = wb_robot_get_device("right wheel motor");
WbDeviceTag left_motor = wb_robot_get_device("left wheel motor");
```

3. Mengatur posisi motor robot yaitu dengan :
- ```
wb_motor_set_position(right_motor, 10.0);
```

Motor yang dijalankan adalah motor sebelah kanan, sehingga menggunakan variabel yang telah dideklarasikan sebelumnya. Jika, menginginkan untuk menggerakkan motor pada sebelah kiri maka ubah variabel dengan left\_motor.

Kemudian, sebelumnya jangan lupa untuk menambahkan library motor yaitu sebagai berikut :

```
#include <webots/robot.h>
#include <webots/motor.h>
```

4. Untuk menjalankan robot, set terlebih dahulu *controller* yang terdapat pada e-puck dengan nama yang sama, yang telah diinputkan.

```
translation -0.0773 -0.337 -1.62e-05
rotation 0.00109 -0.0174 1 0.124
name "e-puck"
controller "gerak_robot_sederhana"
```

5. Posisi robot bergeser setelah melakukan *running*.

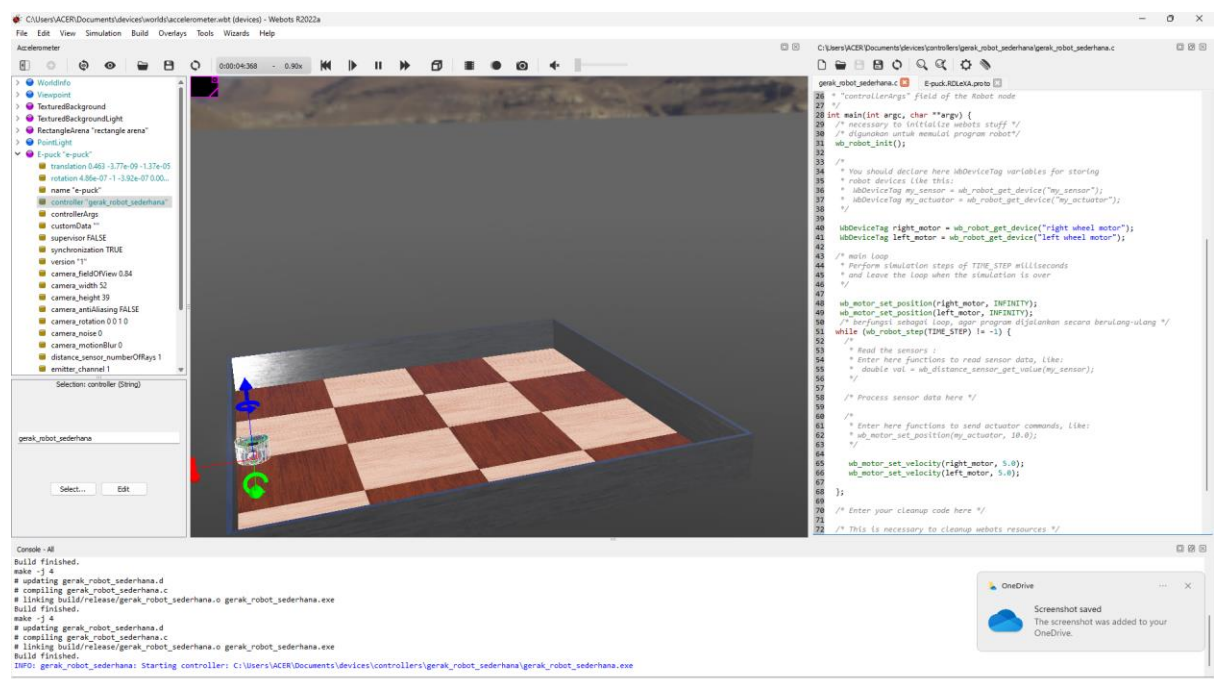
6. Jika, ingin robot bergerak dengan menggunakan kecepatan bukan posisi maka :

```
wb_motor_set_position(right_motor, INFINITY);
wb_motor_set_position(left_motor, INFINITY);
```

INFINITY artinya tidak menggunakan kendali posisi, tapi kecepatan.

Dengan tambahan (didalam *loop*) :

```
wb_motor_set_velocity(right_motor, 5.0);
wb_motor_set_velocity(left_motor, 5.0);
```



Robot akan bergerak sampai ujung.

Jika, dalam pemrograman dirubah seperti ini :

```
wb_motor_set_velocity(right_motor, 5.0);
wb_motor_set_velocity(left_motor, -5.0);
```

Maka, robot akan berputar ditempat :

