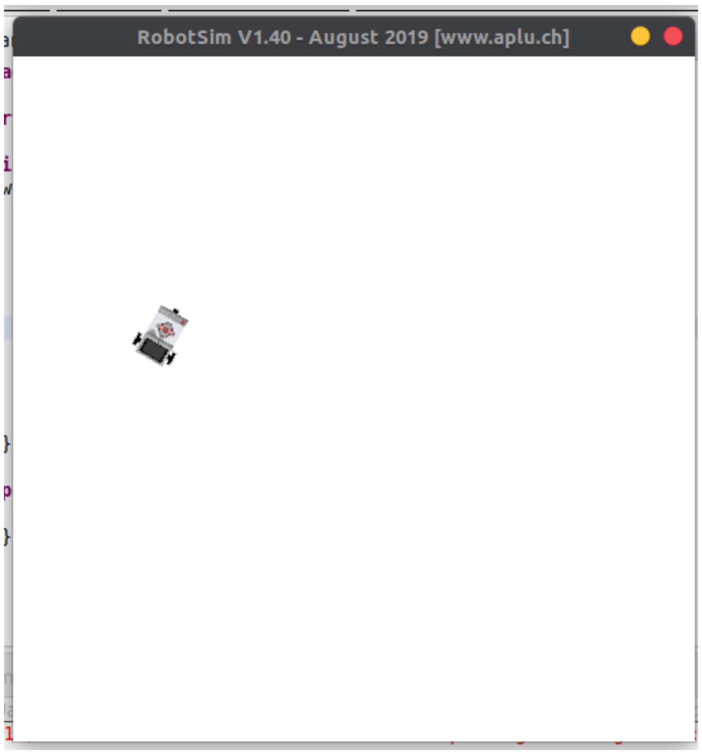
Practical 1

Aim: Write a program to create a robot with gear and move it forward, left, right.

a. withGear.java

```
package practical_1;

import ch.aplu.robotsim.*;

public class withGear {
    withGear(){
        NxtRobot robot = new NxtRobot();
        Gear g = new Gear();
        robot.addPart(g);
        g.setSpeed(100);
        g.forward(800);
        g.left(270);
        g.forward(800);
        g.right(270);
        g.forward(500);
    }
    public static void main (String[] args) {
        withGear robo = new withGear();
    }
}
```

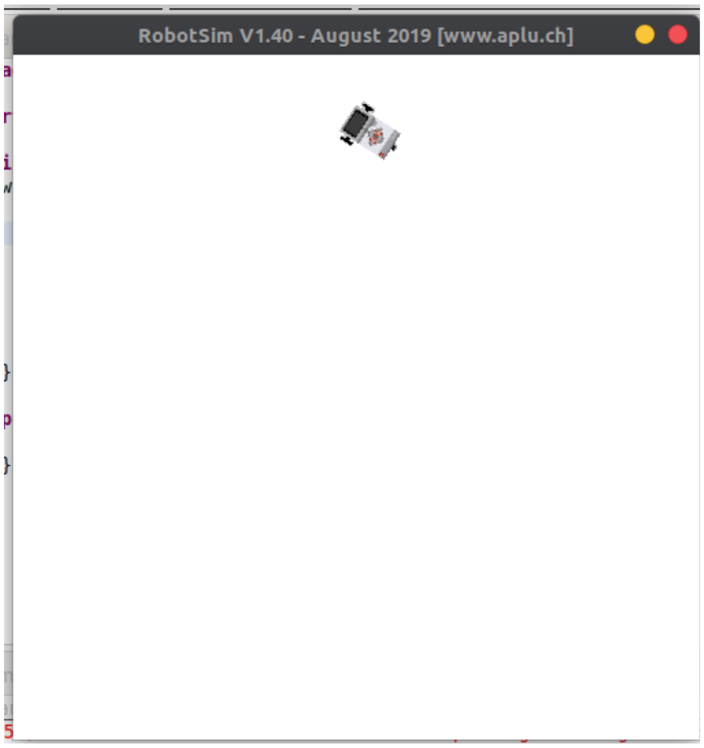
Output:



b. withoutGear.java

```
package practical_1;

import ch.aplu.robotsim.*;

public class withoutGear {
    withoutGear(){
        TurtleRobot robot = new TurtleRobot();
        robot.forward(200);
        robot.left(90);
        robot.forward(200);
        robot.left(90);
        robot.forward(200);
    }
    public static void main (String[] args) {
        withoutGear robo = new withoutGear();
          }
}
```

Output:



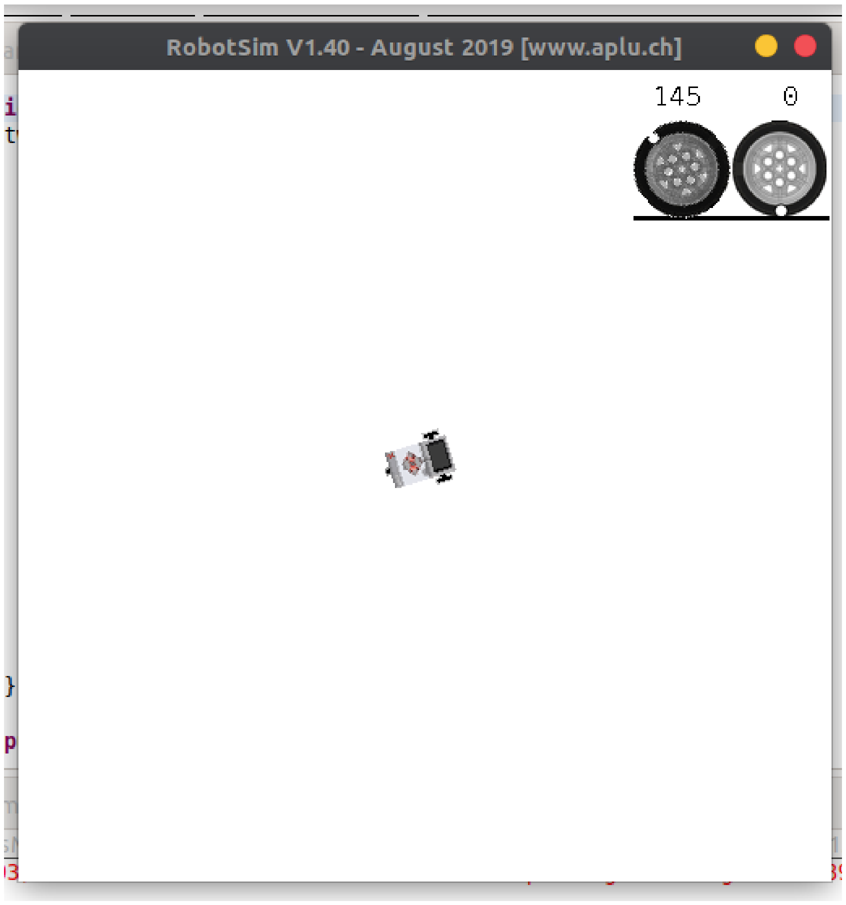
Practical 2

Aim: Write a program to create a robot with 2 motors and move it forward, left, right.

twoMotorsMovement.java

```
package practical_2;

import ch.aplu.robotsim.*;

public class twoMotorsMovement {
    twoMotorsMovement(){
        NxtRobot r = new NxtRobot();
        Motor m1 = new Motor(MotorPort.A);
        Motor m2 = new Motor(MotorPort.B);
        r.addPart(m1);
        r.addPart(m2);

        m1.forward();
        Tools.delay(1090);
        m2.forward();

        Tools.delay(1090);
        m1.stop();

        m2.forward();
        Tools.delay(1090);
        m1.forward();

        m1.stop();
        m2.stop();
    }
    public static void main(String args[]){
        new twoMotorsMovement();
    }
}
```

Output:



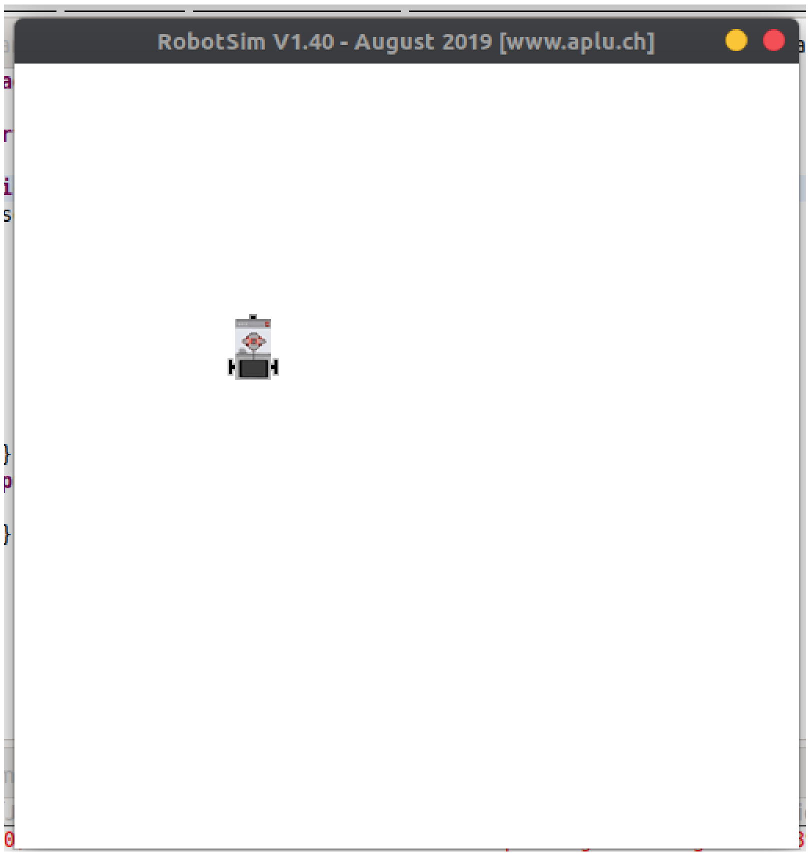
Practical 3

Aim:WRITE A PROGRAM TO DO A SQUARE USING A WHILE LOOP.

square.java

```
package practical_3;

import ch.aplu.robotsim.*;

public class square{
     square(){
          NxtRobot robot = new NxtRobot();
          Gear g = new Gear();
          robot.addPart(g);
          g.setSpeed(100);
          while(true) {
               g.forward(1000);
               g.left(270);
          }
     }
     public static void main(String[] args) {
          square robo = new square();
     }
}
```

Output :



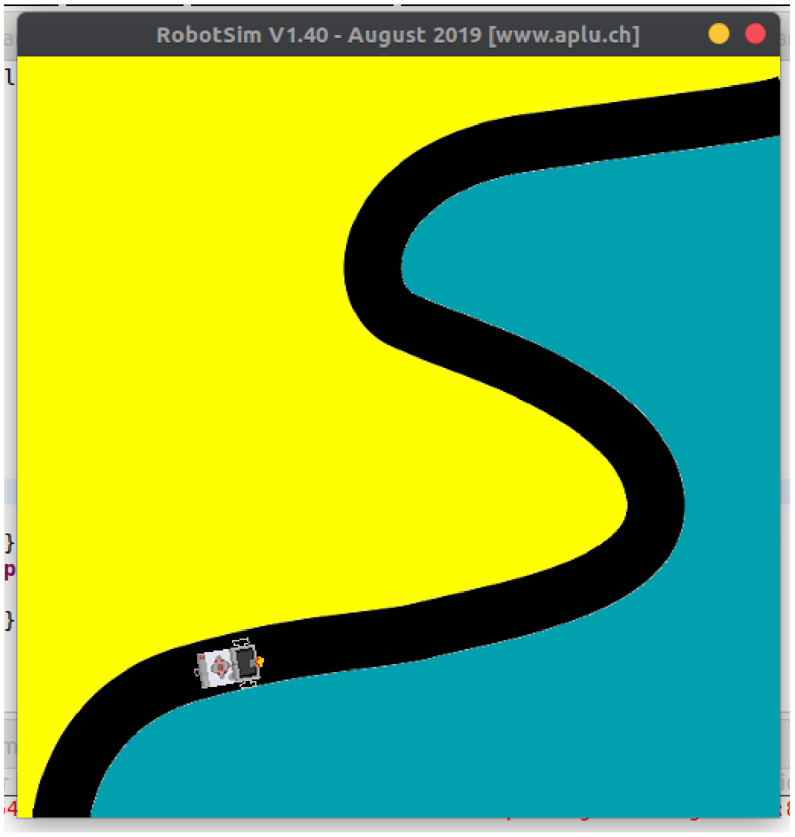
Practical 4

Aim: Write a program to create a robot with light sensors to follow a line.

lightSensor.java

```
package practical_4;

import ch.aplu.robotsim.*;

public class lightSensor {
     static {
        RobotContext.setStartPosition(32,495);
        RobotContext.useBackground("sprites/road.gif");
     }

     lightSensor(){
          LegoRobot robot = new LegoRobot();
          Gear g = new Gear();
          LightSensor ls = new LightSensor(SensorPort.S3);
          robot.addPart(g);
          robot.addPart(ls);
          g.forward();
          g.setSpeed(50);

          while(true){
            int v =ls.getValue();
            if(v < 100)
                g.forward();
            if(v > 350 && v<750)
                g.leftArc(0.005);
            if(v > 800)
                g.rightArc(0.005);
        }
     }
public static void main(String[] args) {
            lightSensor robo = new lightSensor();
      }
}
```

Output :



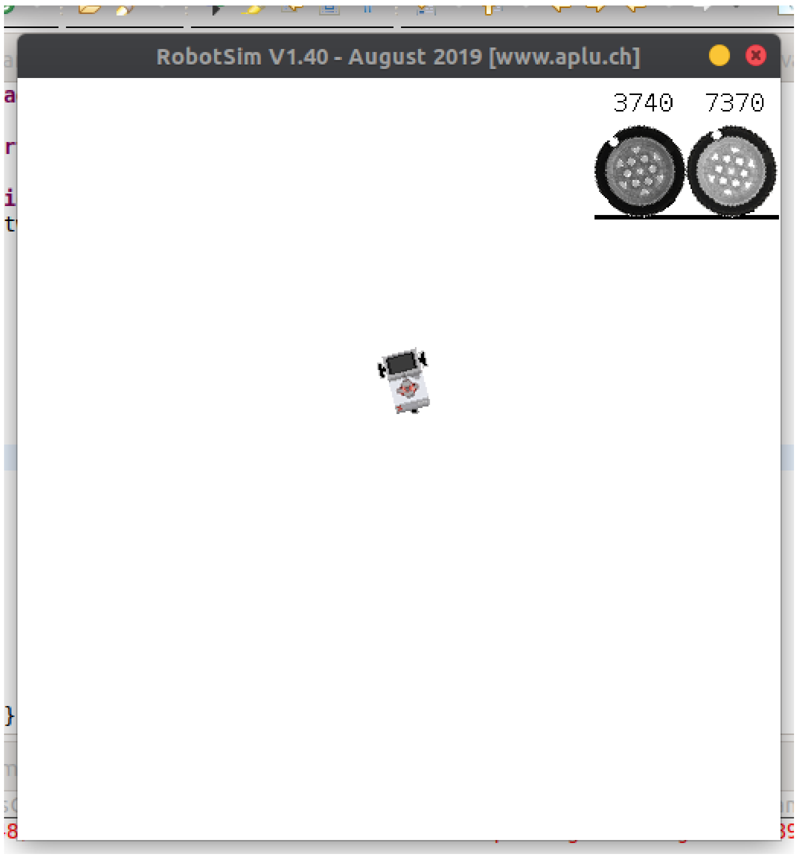
Practical 5

Aim: Write a program to create a robot that does a circle using 2 motors.

twoMotorsCircle.java

```
package practical_5;

import ch.aplu.robotsim.*;

public class twoMotorsCircle {
     twoMotorsCircle(){
          NxtRobot robot = new NxtRobot();
          Motor A = new Motor(MotorPort.A);
          Motor B = new Motor(MotorPort.B);
          robot.addPart(A);
          robot.addPart(B);

          A.setSpeed(100);
          B.setSpeed(100);
          A.forward();
          B.forward();

          while(true) {
               Tools.delay(500);
               A.stop();
               Tools.delay(500);
               A.forward();
          }

     }
     public static void main(String[] args) {
          twoMotorsCircle robo = new twoMotorsCircle();
     }
}
```

<u>Output :</u>



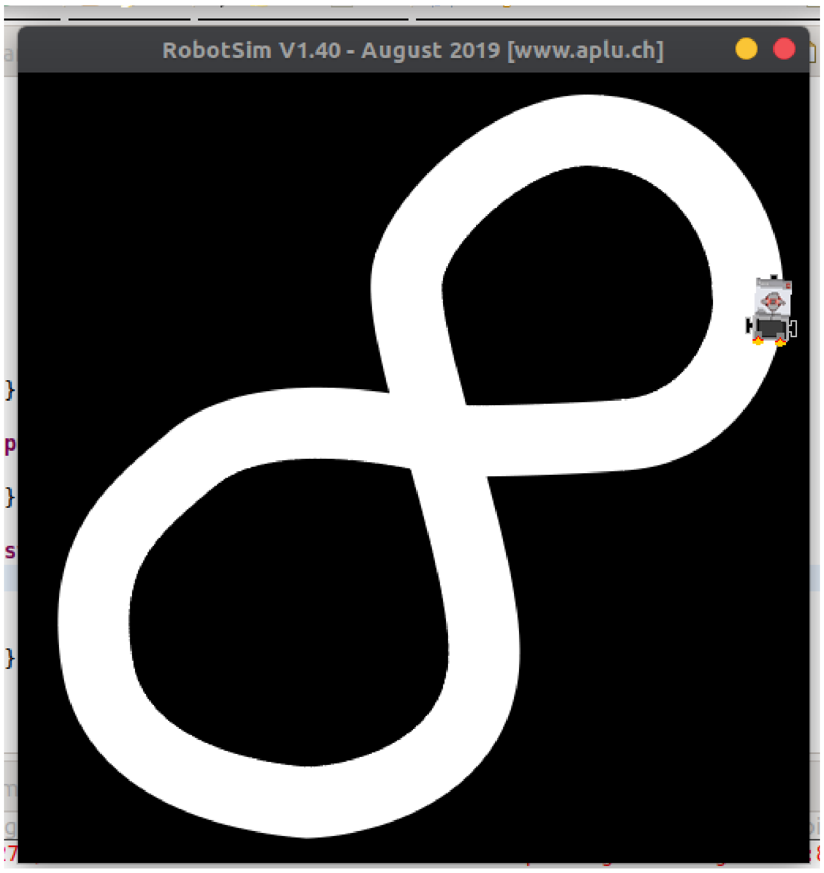
Practical 6

Aim: Write a program to create a path following robot.

pathFollowig.java

```
package practical_6;

import ch.aplu.robotsim.*;
public class pathFinding {
     pathFinding(){
          NxtRobot robot=new NxtRobot();
          Gear gear=new Gear();
          LightSensor ls1=new LightSensor(SensorPort.S1);
          LightSensor ls2=new LightSensor(SensorPort.S2);
          robot.addPart(gear);
          robot.addPart(ls1);
          robot.addPart(ls2);
          gear.forward();
          gear.setSpeed(100);

          while(true) {
           int rightValue=ls1.getValue();
           int leftValue=ls2.getValue();
           if(leftValue < 10)
               gear.rightArc(0.05);
           if(rightValue < 10)
               gear.leftArc(0.05);
           if(leftValue > 10 && rightValue > 10)
               gear.forward();
          }

     }
public static void main(String[] args) {
          pathFinding robot = new pathFinding();
     }

     static {
        NxtContext.setStartPosition(267,232);
        NxtContext.setStartDirection(-90);
        NxtContext.useBackground("sprites/path.gif");
     }
}
```

Output :



Practical 7

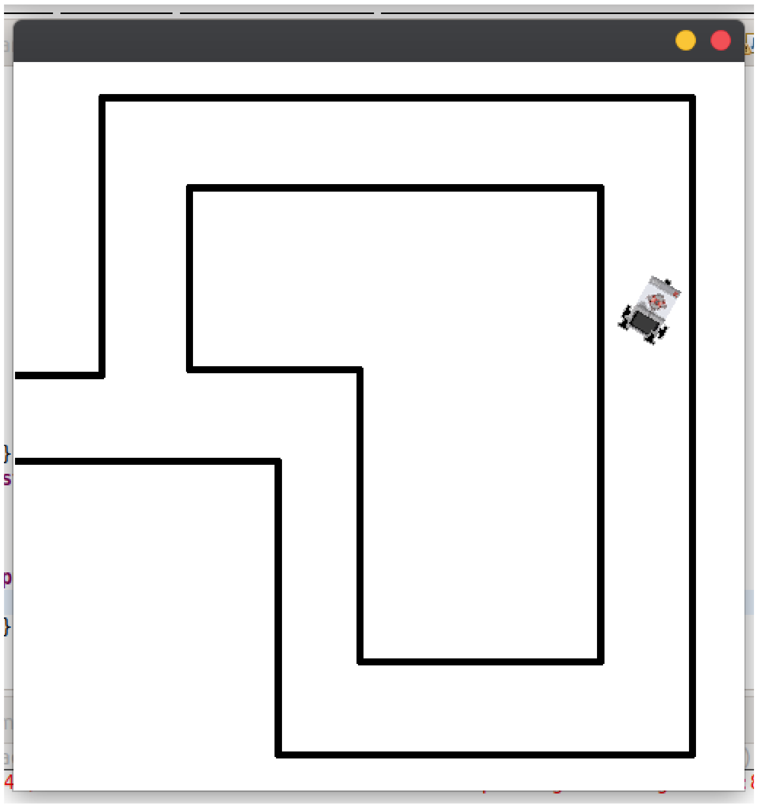
Aim: Write a program to resist obstacles.

resistObstacle.java

```
package practical_7;

import ch.aplu.robotsim.*;

public class resistObstacle {
     resistObstacle(){
        LegoRobot r=new LegoRobot();
        Gear g = new Gear();
        TouchSensor t1= new TouchSensor(SensorPort.S1);
        TouchSensor t2 = new TouchSensor(SensorPort.S2);
        r.addPart(g);
        r.addPart(t1);
        r.addPart(t2);
        g.forward();
        g.setSpeed(50);

        while(true){
            Boolean b1 = t1.isPressed();
            Boolean b2 = t2.isPressed();
            if(b1 && b2){
                g.backward(150);
                g.right(400);
                g.forward();
            }
            if(b1){
                g.backward(150);
                g.left(200);
                g.forward();
            }
```

```
                if(b2){
                    g.backward(150);
                    g.right(200);
                    g.forward();
                }
            }
        }
    static {
        RobotContext.setStartPosition(100,250);
        RobotContext.useObstacle(RobotContext.channel);
     }
    public static void main(String[] args) {
     resistObstacle robo = new resistObstacle();
    }
}
```

Output :



Practical 8

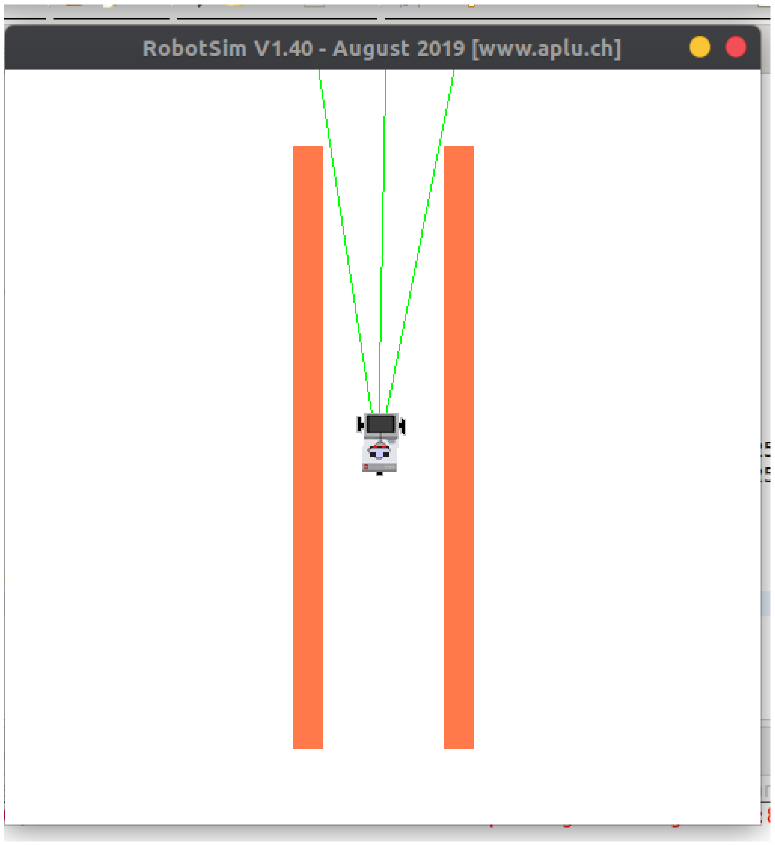
Aim: Ultrasonic sensor

ultrasonicSensor.java

```
package practical_8;

import java.awt.Color;
import java.awt.Point;

import ch.aplu.robotsim.*;

public class ultrasonicSensor {
     ultrasonicSensor(){
          LegoRobot robot = new LegoRobot();
        Gear gear = new Gear();
        robot.addPart(gear);
        UltrasonicSensor us = new
UltrasonicSensor(SensorPort.S1);
        robot.addPart(us);
        us.setBeamAreaColor(Color.green);
        us.setProximityCircleColor(Color.lightGray);

        double arc = 0.5;
        gear.setSpeed(50);
        gear.rightArc(arc);
        boolean isRightArc = true;

        int oldDistance = 0;
        while(true) {
          Tools.delay(100);
            int distance = us.getDistance();
            if (distance == -1)
```

```
               continue;
            if (distance < oldDistance)
            {
              if (isRightArc)
              {
                gear.leftArc(arc);
                isRightArc = false;
              }
              else
              {
                gear.rightArc(arc);
                isRightArc = true;
              }
            }
            oldDistance = distance;
        }
     }

     static {
           Point[] mesh_bar =
                  {
                    new Point(10, 200), new Point(-10, 200),
                    new Point(-10, -200), new Point(10, -200)
                  };
           RobotContext.useTarget("sprites/bar1.gif", mesh_bar,
200, 250);
         RobotContext.useTarget("sprites/bar1.gif", mesh_bar,
300, 250);
         RobotContext.setStartPosition(250, 460);
     }

     public static void main(String[] args) {
          ultrasonicSensor robo = new ultrasonicSensor();
     }
}
```

<u>Output :</u>


RobotSim V1.40 - August 2019 [www.aplu.ch]