Robotics Journal

M.Sc Part II Computer Science

Shubham Lad 502

October 13, 2023

R.J. College of Arts, Science & Commerce
Robotics
Seat number: 502

Contents

List of Figures

Practical No: 1

Aim:A) Moving a robot left, right, and forward using gears.

Code:

```java
package robotics;
import ch.aplu.robotsim.Gear;
import ch.aplu.robotsim.NxtRobot;
/**
 *
 * @author
 */

public class Robo_p1a {
    /**
     * @param args the command line arguments
     */
    Robo_p1a(){
        NxtRobot robot=new NxtRobot();

        Gear gear=new Gear();
        robot.addPart(gear);

        gear.setSpeed(10);

        gear.forward(4000);
        gear.left(5000);

        gear.forward(5000);
        gear.right(4800);

        robot.exit();
    }
    public static void main(String[] args) {
        // TODO code application logic here
        Robo_p1a m=new Robo_p1a();
    }

}
```

Output :

Aim:B) Moving a robot in a square motion.

Code:

```
package robotics;
import ch.aplu.robotsim.TurtleRobot;
/**
 *
 * @author mithunparab
 */

public class Robo_p1b {
    Robo_p1b()
     {
     TurtleRobot t = new TurtleRobot();
     t.forward(100);
     t.right(30);
     t.forward(50);
     t.left(90);
     t.forward(70);
     }
     public static void main(String[] args){
     new Robo_p1b();
    }
}
```

Output :

Practical No: 2

Aim: Robot performing circular motion

Code:

```
package robotics;

/**
 *
 * @author
 */
import ch.aplu.robotsim.Gear;
import ch.aplu.robotsim.NxtRobot;

public class Robo_p2 {

    Robo_p2(){
        NxtRobot robot=new NxtRobot();

        Gear gear=new Gear();
        robot.addPart(gear);
        gear.setSpeed(150);

        for (int i=0;i<100;i++){
            gear.forward(150);
            gear.right(30);
        }
        robot.exit();
    }

    public static void main(String[] args) {
        // TODO code application logic here
        Robo_p2 m=new Robo_p2();
    }
}
```

Output :

Practical No: 3

Aim: Moving a robot with motors

Code:

```
ackage robotics;


import ch.aplu.robotsim.Motor;
import ch.aplu.robotsim.MotorPort;
import ch.aplu.robotsim.NxtRobot;
import ch.aplu.robotsim.Tools;
/**
 *
 * @author admin
 */


//Aim: Move robot with the help of motors
public class Robo_p3 {
    public Robo_p3(){
        NxtRobot robot = new NxtRobot();
        Motor motA = new Motor(MotorPort.A);
        Motor motB = new Motor(MotorPort.B);
        robot.addPart(motA);
        robot.addPart(motB);
        motA.forward();
        motB.forward();
```

```
        Tools.delay(2000);

        motA.stop();
        Tools.delay(1050);
        motA.forward();
        Tools.delay(2000);

        motB.stop();
        Tools.delay(1050);
        motB.forward();
        Tools.delay(2000);

        robot.exit();
    }

    public static void main(String args[]){
        Robo_p3 r = new Robo_p3();
    }
}
```

Output :

Practical No: 4

Aim: A) Circular motion using arch function with gear.

Code:

```
package robotics;

import ch.aplu.robotsim.Gear;
import ch.aplu.robotsim.NxtRobot;
import ch.aplu.robotsim.Tools;
/**
 *
 * @author admin
 */
public class Robo_p4a {

    Robo_p4a(){
        NxtRobot robot=new NxtRobot();
        Gear gear = new Gear();
        robot.addPart(gear);
        gear.setSpeed(60);
        gear.leftArc(0.2,7000);
        gear.rightArc(0.2);
        Tools.delay(5000);
        robot.exit();
    }

    public static void main(String args[]){
        Robo_p4a p4a=new Robo_p4a();
    }
}
```

Output :

Aim: B) Circular motion using arch function without gear.

Code:

```java
package robotics;

import ch.aplu.robotsim.*;
import ch.aplu.robotsim.NxtRobot;
/**
 *
 * @author admin
 */
public class Robo_p4b {
    Robo_p4b(){
        NxtRobot robot=new NxtRobot();
        Gear gear = new Gear();
        robot.addPart(gear);
        gear.forward(200);
        gear.setSpeed(60);
        gear.leftArc(0.2,7000);
        gear.forward(200);
        gear.leftArc(0.2,7000);
        gear.forward(200);
        gear.leftArc(0.2,7000);
        gear.forward(200);
        gear.leftArc(0.2,7000);
        gear.forward(200);
        robot.exit();
    }
    public static void main(String args[]){
        Robo_p4b p4b=new Robo_p4b();
        NxtContext.setStartPosition(250,200);
```

```
29          NxtContext.setStartDirection(90);
30      }
31  }
```

Output :

Practical No: 5

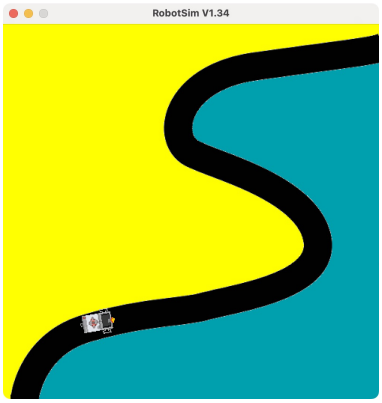
Aim: A) Line-following robot.

Code :

```
package robotics;

import ch.aplu.robotsim.*;

/**
 *
 * @author admin
 */
public class Robo_p5a {
    Robo_p5a(){
        LegoRobot robot=new LegoRobot();
        Gear gear = new Gear();
        LightSensor ls = new LightSensor(SensorPort.S3);
        robot.addPart(gear);
        gear.setSpeed(20);
        robot.addPart(ls);
        while (true){
        int v=ls.getValue();
        if (v<100)//black
            gear.forward();
        if (v>300 && v<750)//blue
            gear.leftArc(0.05);
        if (v>800)//yellow
            gear.rightArc(0.05);
    }
    }
    public static void main(String args[]){
        Robo_p5a p5a = new Robo_p5a();
    }

    static{
        RobotContext.setStartPosition(50, 490);
        RobotContext.setStartDirection(-90);
        RobotContext.useBackground("sprites/road.gif");
    }
```

```
}
```

Output :



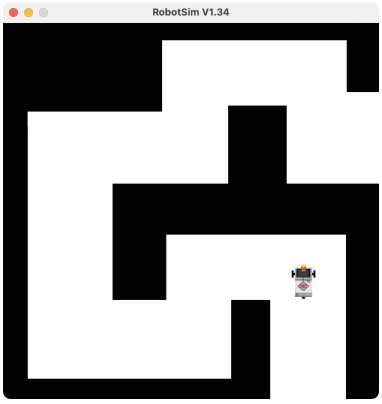
Aim: B) Line-following robot on a different image.

Code :

```
package robotics;

import ch.aplu.robotsim.Gear;
import ch.aplu.robotsim.LegoRobot;
import ch.aplu.robotsim.LightSensor;
import ch.aplu.robotsim.RobotContext;
import ch.aplu.robotsim.SensorPort;

/**
 *
 * @author admin
 */
public class Robo_p5b {
    Robo_p5b(){
        LegoRobot robot=new LegoRobot();
        Gear gear = new Gear();
        LightSensor ls = new LightSensor(SensorPort.S3);
        robot.addPart(gear);
        gear.setSpeed(50);
```

```
        robot.addPart(ls);
        while (true){
        int v=ls.getValue();
        System.out.println(v);
       gear.forward();
        if (v>1000 || v<=7)
            gear.leftArc(0.05);

      //  for (int i=0;i>10000;i++){
            //gear.forward(150);
         //   gear.right(30);
       // }
         //   gear.rightArc(0.05);
        }
    }
    public static void main(String args[]){
        Robo_p5b p5b = new Robo_p5b();
    }
    static{
        RobotContext.setStartPosition(400, 450);
        //RobotContext.setStartDirection(-90);
        RobotContext.useBackground("sprites/bg2.gif");
    }
}
```

Output :



Practical No: 6

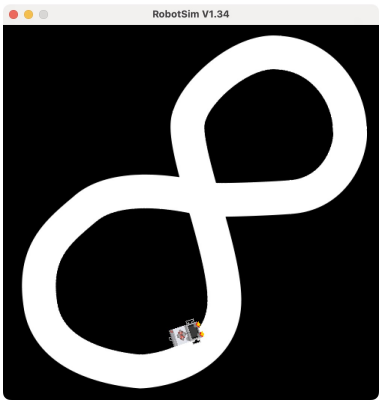
Aim: A) Path finding robot.

Code :

```
package robotics;

import ch.aplu.robotsim.*;
import ch.aplu.robotsim.LightSensor;
import ch.aplu.robotsim.NxtRobot;
import ch.aplu.robotsim.SensorPort;

/**
 *
 * @author admin
 */
public class Robo_p6a {
    Robo_p6a(){

        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();

        while(true)
        {
            int rightValue = ls1.getValue();
            int leftValue = ls2.getValue();
            int d = rightValue-leftValue;

            if (d>100)
                gear.rightArc(0.1);
            if (d< -100)
                gear.leftArc(0.1);
            if (d>-100 && d<100 && rightValue>500)
                gear.forward();
```

```
        }
    }

    public static void main(String args[]){
        Robo_p6a p6a=new Robo_p6a();
    }
    static{
        NxtContext.setStartPosition(250,490);
        NxtContext.setStartDirection(-90);
        NxtContext.useBackground("sprites/path.gif");
    }
}
```

Output :



Aim: B) Path finding robot on a different image

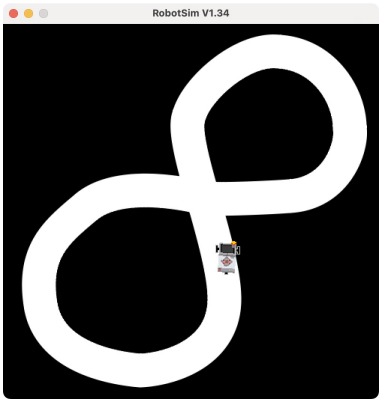
Code :

```
package robotics;

import ch.aplu.robotsim.Gear;
import ch.aplu.robotsim.LightSensor;
import ch.aplu.robotsim.NxtContext;
import ch.aplu.robotsim.NxtRobot;
import ch.aplu.robotsim.SensorPort;

```

```java
/**
 *
 * @author admin
 */
public class Robo_p6b {

    Robo_p6b(){
        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();

        while(true)
        {
            int d = ls1.getValue();
            //System.out.print(d+'\n');
            //int leftValue = ls2.getValue();
            //int d = rightValue-leftValue;

            if (d>100)
                gear.rightArc(0.1);
            if (d<100)
                //gear.forward();
                gear.leftArc(0.1);
            if (d>100 && d<-100)
                gear.forward();
        }
    }

    public static void main(String args[]){
        Robo_p6b p6b=new Robo_p6b();
    }
    static{
        NxtContext.setStartPosition(250,490);
        NxtContext.setStartDirection(-90);
        NxtContext.useBackground("sprites/path.gif");
    }
```

```
50 }
```

Output :



Practical No: 7

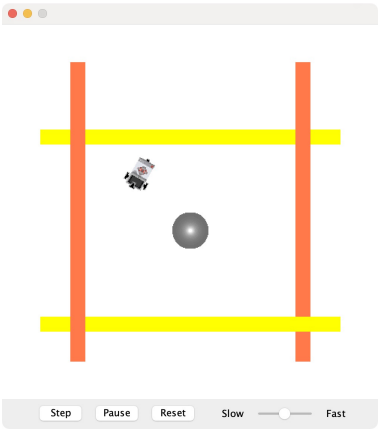
Aim: Obstacle resistance using touch sensor.

Code :

```
package robotics;
import ch.aplu.robotsim.*;

public class Robo_p7 {

    static {
        RobotContext.showNavigationBar();
        RobotContext.useObstacle("sprites/bar2.gif", 250, 150); //up
        RobotContext.useObstacle("sprites/bar1.gif", 400, 250); //right
        RobotContext.useObstacle("sprites/torch.png", 250, 275); //center circle
        RobotContext.useObstacle("sprites/bar2.gif", 250, 400); //down
        RobotContext.useObstacle("sprites/bar1.gif", 100, 250); //left
    }

    public Robo_p7() {
        LegoRobot robot = new LegoRobot();
        Gear gear = new Gear();
        TouchSensor ts = new TouchSensor(SensorPort.S3);
        robot.addPart(gear);
        robot.addPart(ts);
        gear.setSpeed(30);
        gear.forward();

        while (true) {
            if (ts.isPressed()) {
                gear.backward(500);
                gear.left(750);
                gear.forward();
            }
        }
    }

    public static void main(String[] args) {
        Robo_p7 mp = new Robo_p7();
    }
```

```
36
37 }
```

Output :



Practical No: 8

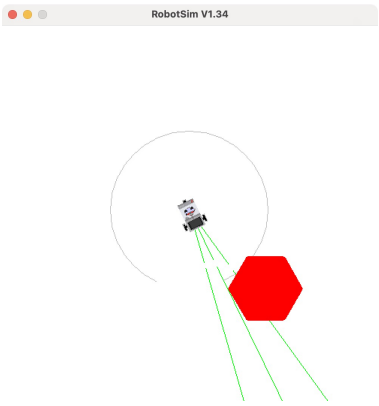
Aim: Write a program to create a robot with ultrasonic sensor.

Code :

```
/*
Search fixed target using the ultrasonic Sensor
 */
package robotics;

import ch.aplu.robotsim.*;
import java.awt.*;

public class Robo_p8 {
    private final LegoRobot robot;
    private final Gear gear;
    private final UltrasonicSensor us;

    public Robo_p8() {
        robot = new TurtleRobot();
        gear = new Gear();
        robot.addPart(gear);
        gear.setSpeed(10);
        us = new UltrasonicSensor(SensorPort.S1);
        robot.addPart(us);
        us.setBeamAreaColor(Color.green);  // May be commented out
        us.setProximityCircleColor(Color.lightGray); // May be commented out
        runRobotProgram();
    }

    private void runRobotProgram() {
        searchTarget();
        while (true) {
            if (us.getDistance() < 10)
                gear.stop();
        }
    }

    private void searchTarget() {
        while (true) {
```

```
            gear.right(50);
            int distance = us.getDistance();
            if (distance != -1) {
                gear.right(1500);
                gear.forward();
                return;
            }
        }
    }

    public static void main(String[] args) {
        Robo_p8 a = new Robo_p8();
    }

// ------------------ Environment --------------------------
    static {
        Point[] mesh = {
            new Point(50, 0), new Point(25, 42), new Point(-25, 42),
            new Point(-50, 0), new Point(-25, -42), new Point(25, -42)
        };

        RobotContext.useTarget("sprites/target_red.gif", mesh, 350, 350);
    }
}
```

Output :



Practical No: 9

Aim: Torch follower.

Code :

```
package robotics;
import ch.aplu.robotsim.*;
/**
 *
 * @author
 */

public class Robo_p9
{
  public Robo_p9()
  {
    LegoRobot robot = new LegoRobot();
    LightSensor lsFR = new LightSensor(SensorPort.S1, true);
    LightSensor lsFL = new LightSensor(SensorPort.S2, true);
    LightSensor lsRR = new LightSensor(SensorPort.S3, true);
    LightSensor lsRL = new LightSensor(SensorPort.S4, true);

    Gear gear = new Gear();
    robot.addPart(gear);
    robot.addPart(lsFR);
    robot.addPart(lsFL);
    robot.addPart(lsRL);
    robot.addPart(lsRR);

    gear.setSpeed(25);
    gear.forward();
    double s = 0.02;
    while (!robot.isEscapeHit())
    {
      int vFR = lsFR.getValue();
      int vFL = lsFL.getValue();
      int vRR = lsRR.getValue();
      int vRL = lsRL.getValue();
      double d = 1.0 * (vFL - vFR) / (vFL + vFR);

```

```
      if (vRL + vRR > vFL + vFR)  // torch behind robot
        gear.left();
      else if (d > -s && d < s)
        gear.forward();
      else
      {
        if (d >= s)
          gear.leftArc(0.05);
        else
          gear.rightArc(0.05);
      }
      Tools.delay(100);
    }
    robot.exit();
  }

  public static void main(String[] args)
  {
    new Robo_p9();
  }

  // ------------------ Environment -------------------------
  static
  {
    RobotContext.useTorch(1, 150, 250, 100);
  }
}
```

Output :

Practical No: 10

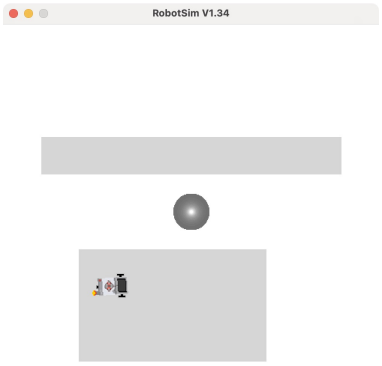
Aim: Shadow follower.

Code :

```
package robotics;
import ch.aplu.robotsim.*;
import java.awt.Color;
/**
 *
 * @author
 */

public class Robo_p10
{
  public Robo_p10()
  {
    LegoRobot robot = new LegoRobot();
    LightSensor ls = new LightSensor(SensorPort.S3, true);
    robot.addPart(ls);
    Gear gear = new Gear();
    robot.addPart(gear);

    gear.leftArc(0.5);
    boolean isLightOn = false;
    while (!robot.isEscapeHit())
    {
      int v = ls.getValue();
```

```
24        if (!isLightOn && v == 0)
25        {
26          isLightOn = true;
27          robot.playTone(2000, 100);
28        }
29        if (isLightOn && v > 0)
30        {
31          isLightOn = false;
32          robot.playTone(1000, 100);
33        }
34        Tools.delay(100);
35      }
36      robot.exit();
37    }
38
39    public static void main(String[] args)
40    {
41      new Robo_p10();
42    }
43
44    // ------------------ Environment --------------------------
45    static
46    {
47      RobotContext.useTorch(1, 250, 250, 100);
48      RobotContext.useShadow(50, 150, 450, 200);
49      RobotContext.useShadow(100, 300, 350, 450);
50    }
51  }
```

Output :


RobotSim V1.34