­­­­­Roll No : 518

Name: Sreejith Sathyadevan

MSc CS Part 2

**Robotics Assignment**

Assignment 1(A):

Aim: Write a program to create a robot to perform rectangular motion using gears

Description:

1] NxtRobot() :

Class that represents a simulated NXT robot brick. Parts (e.g. motors, sensors) may be assembled into the robot to make it doing the desired job.

2] Gear() :

Creates a gear instance with right motor plugged into port A, left motor plugged into port B.

3] addPart() :

Assembles the given part into the robot.

4] setSpeed() :

Sets the speed to the given value (arbitrary units).

5] forward() :

Starts the forward movement for the given duration (in ms) and stops. Method returns at the end of the given duration.

6] left() :

Starts to rotate left (center of rotation at middle of the wheel axes). Method returns immediately, while the movement continues

Code:

import ch.aplu.robotsim.NxtRobot;

import ch.aplu.robotsim.Gear;

public class assignment1A {

public assignment1A() {

NxtRobot r = new NxtRobot ();

Gear g = new Gear();

r.addPart (g);

g.setSpeed (100);

while (true){

g.forward (800);

g.left (280);

}

}

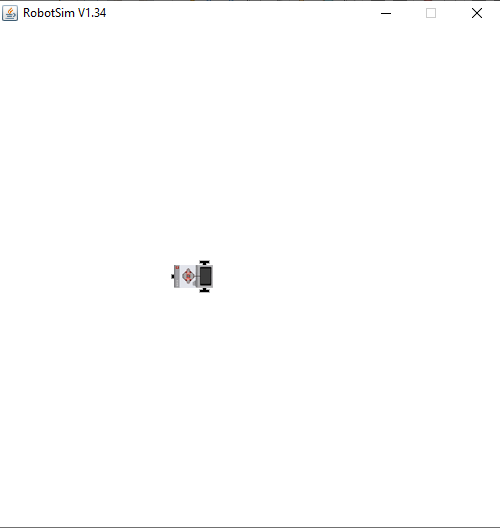
public static void main (String [] args){

new assignment1A ();

}

}

Output:



Assignment 1(B):

Aim: Write a program to create a robot to perform circular motion using gears

Description:

1] rightArc() :

Starts to move to the right on arc with given radius. Method returns immediately, while the movement continues.

.

Code:

import ch.aplu.robotsim.NxtRobot;

import ch.aplu.robotsim.Gear;

public class assignment1B {

public assignment1B () {

NxtRobot r = new NxtRobot ();

Gear g = new Gear ();

r.addPart (g);

g.setSpeed (100);

while (true) {

g.rightArc (0.5);

}

}

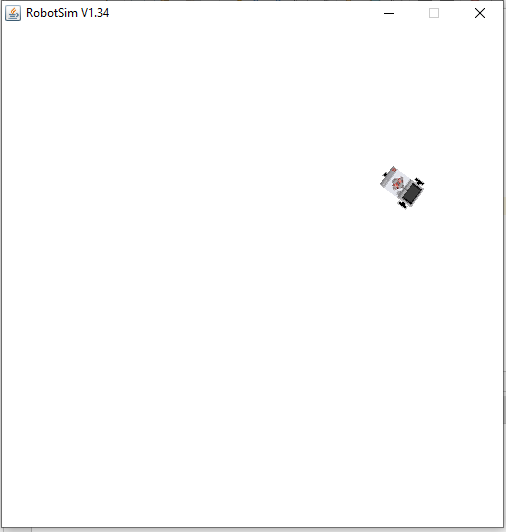
public static void main (String [] args){

new assignment1B ();

}

}

Output:



Assignment 2:

Aim: Write a program to do a square using while or for loop, change direction based on condition and control motor movement

Description:

1] Motor() :

Creates a motor instance that is plugged into given port.

2] Tools.delay() :

Suspends execution of the current thread for the given amount of time.

Code:

import ch.aplu.robotsim.\*;

import java.util.\*;

public class assignment2 {

assignment2 () {

Scanner sc = new Scanner (System.in);

NxtRobot r = new NxtRobot ();

Motor m1 = new Motor (MotorPort.A);

Motor m2 = new Motor (MotorPort.B);

r.addPart (m1);

r.addPart (m2);

System.out.println ("Enter 1 for left and 2 for right :");

int direction = sc.nextInt ();

switch (direction) {

case 1:

for (int i=0; i<4; i++){

m1.forward ();

Tools.delay (1090);

m2.forward ();

Tools.delay (1090);

m1.stop ();

m2.stop ();

}

break;

case 2:

for (int i=0; i<4; i++){

m2.forward ();

Tools.delay (1090);

m1.forward ();

Tools.delay (1090);

m1.stop ();

m2.stop ();

}

break;

}

}

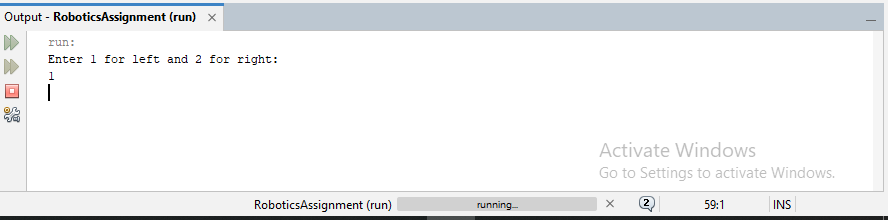
public static void main (String args[]){

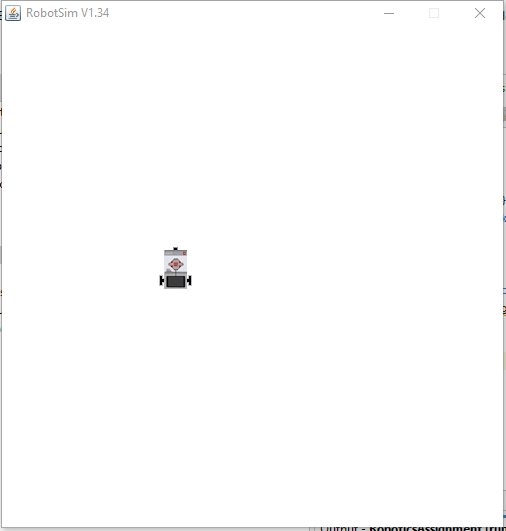
new assignment2 ();

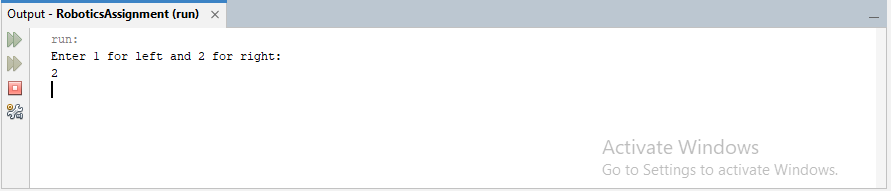
}

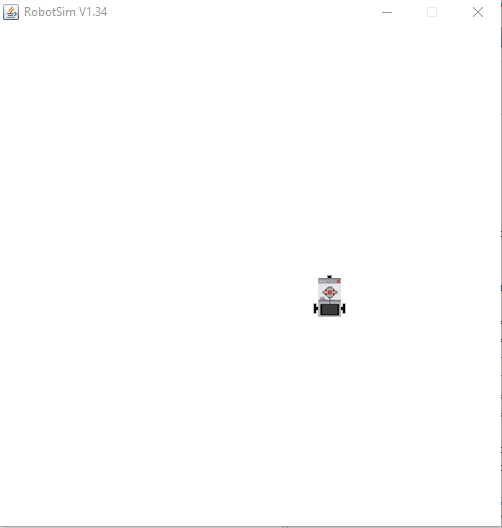
}

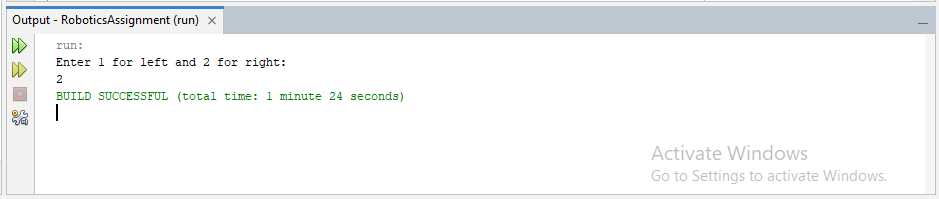
Output:











Assignment 3:

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

Description:

1] LightSensor() :

Creates a sensor instance pointing downwards connected to the given port.

2] leftArc():

Starts to move to the left on arc with given radius. Method returns immediately, while the movement continues.

Code:

import ch.aplu.robotsim.RobotContext;

import ch.aplu.robotsim.Part;

import ch.aplu.robotsim.LightSensor;

import ch.aplu.robotsim.SensorPort;

import ch.aplu.robotsim.Gear;

import ch.aplu.robotsim.LegoRobot;

public class assignment3

{

assignment3 () {

final LegoRobot r = new LegoRobot ();

final Gear g = new Gear ();

final LightSensor l1 = new LightSensor (SensorPort.S1);

final LightSensor l2 = new LightSensor (SensorPort.S2);

r.addPart ((Part) g);

r.addPart ((Part) l1);

r.addPart ((Part) l2);

g.forward ();

g.setSpeed (100);

while (true) {

final int rightValue = l1.getValue ();

final int leftValue = l2.getValue ();

if (leftValue < 10) {

g.rightArc (0.05);

}

else if (rightValue < 10) {

g.leftArc (0.05);

}

if (leftValue > 10 && rightValue > 10) {

g.forward();

}

}

}

public static void main (final String [] args) {

new assignment3 ();

}

static {

RobotContext.setStartPosition (80, 440);

RobotContext.useBackground ("sprites/path.gif");

}

}

Output:

