

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.

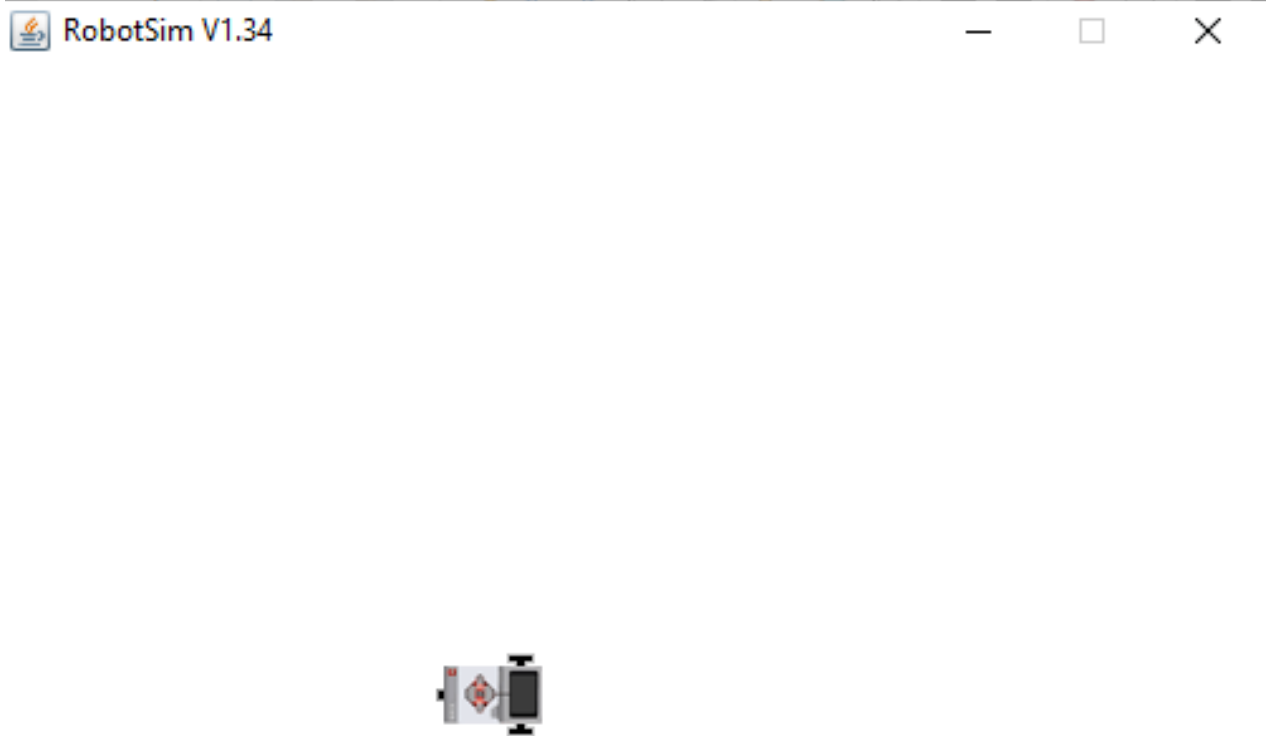
6] left() :

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:

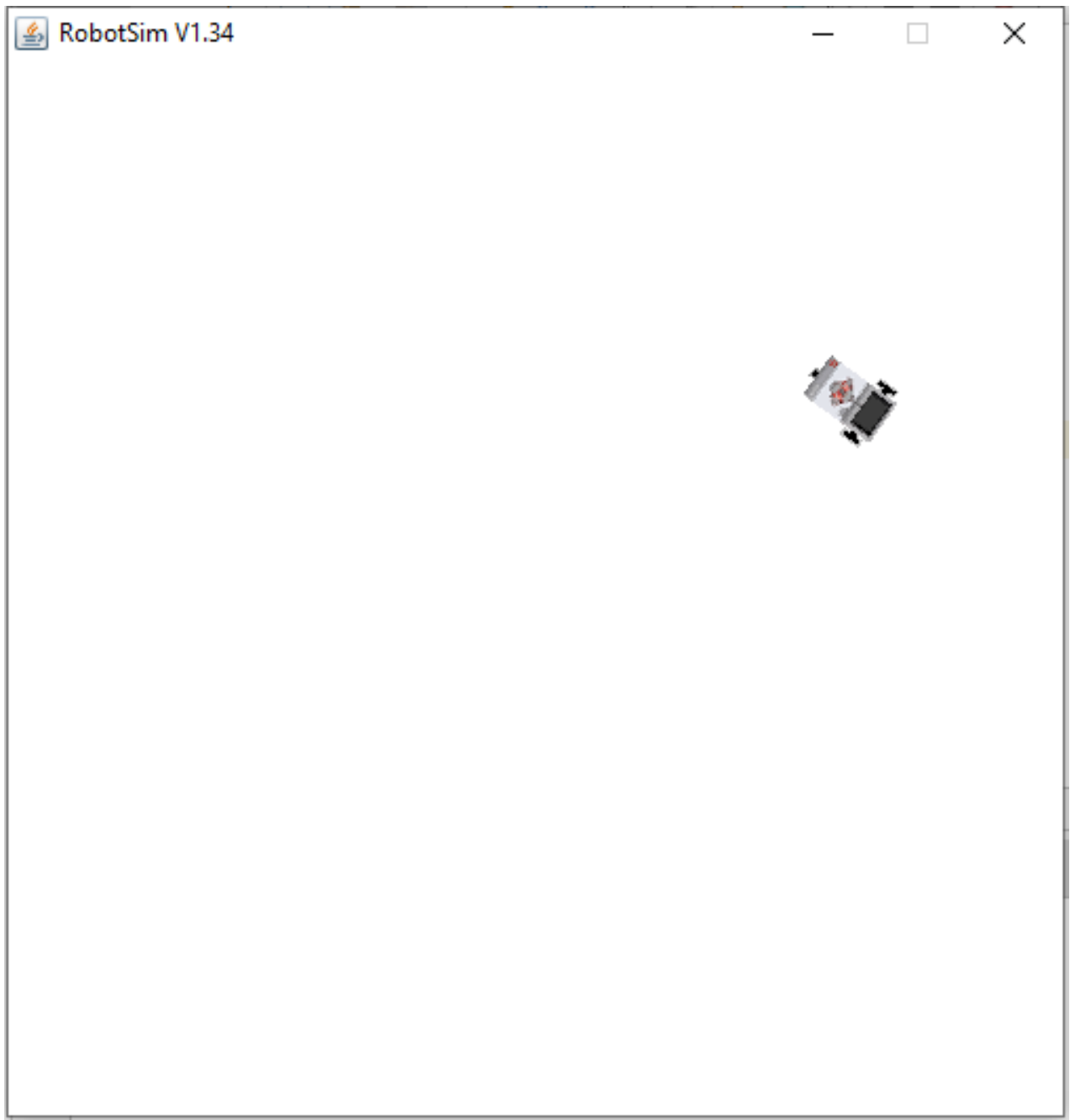
1] rightArc() :

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 (A):

Aim: Write a program to create robot to perform a square motion without using gear.

Code:

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

public class Assignment_2a {

    Assignment_2a () {

        TurtleRobot t = new TurtleRobot ();

        t.setTurtleSpeed (100);

        while (true){

            t.forward(200);

            t.left (90);

        }

    }

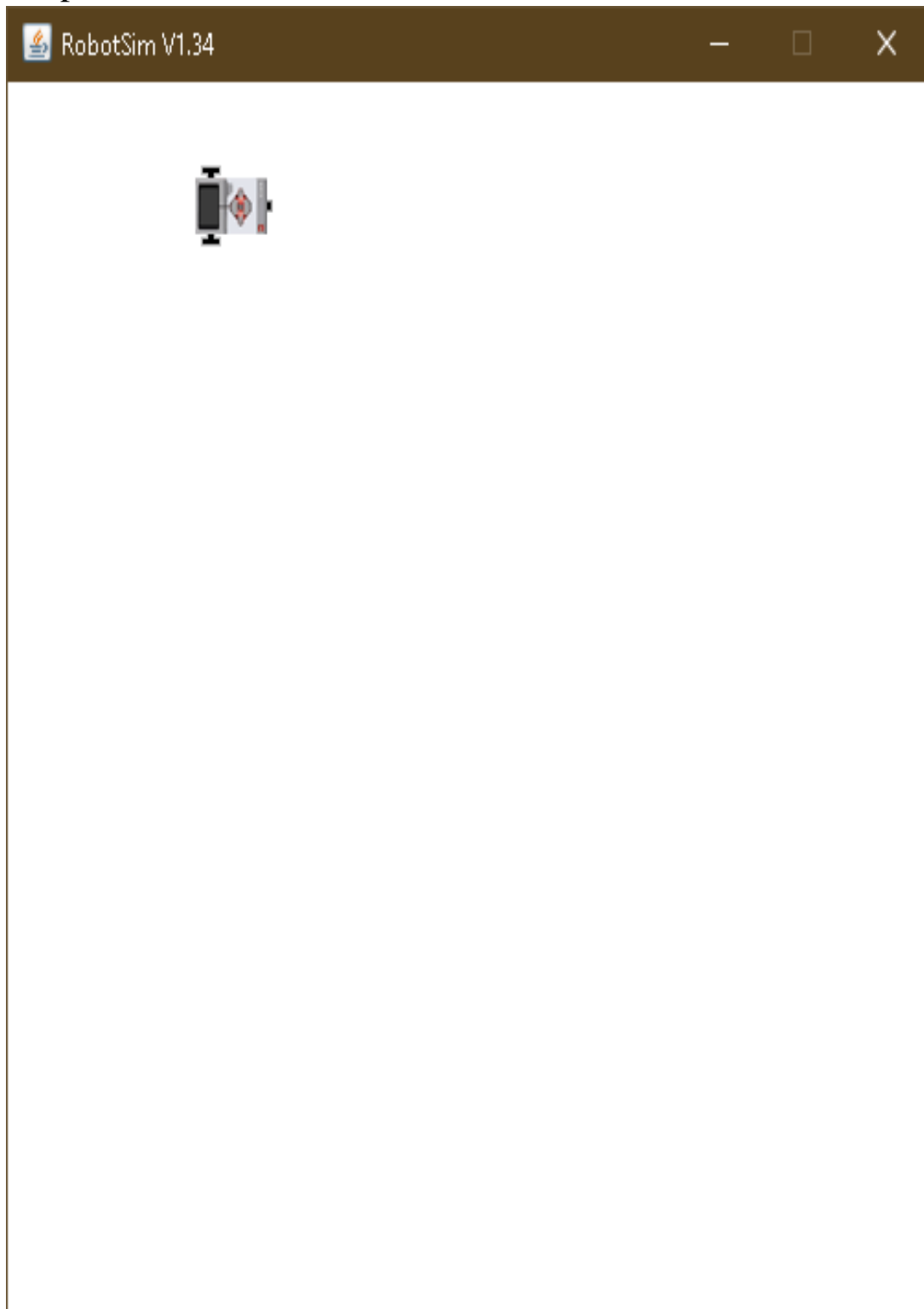
    public static void main (String [] args) {

        new Assignment_1a ();

    }

}
```

Output:



Assignment 2 (B):

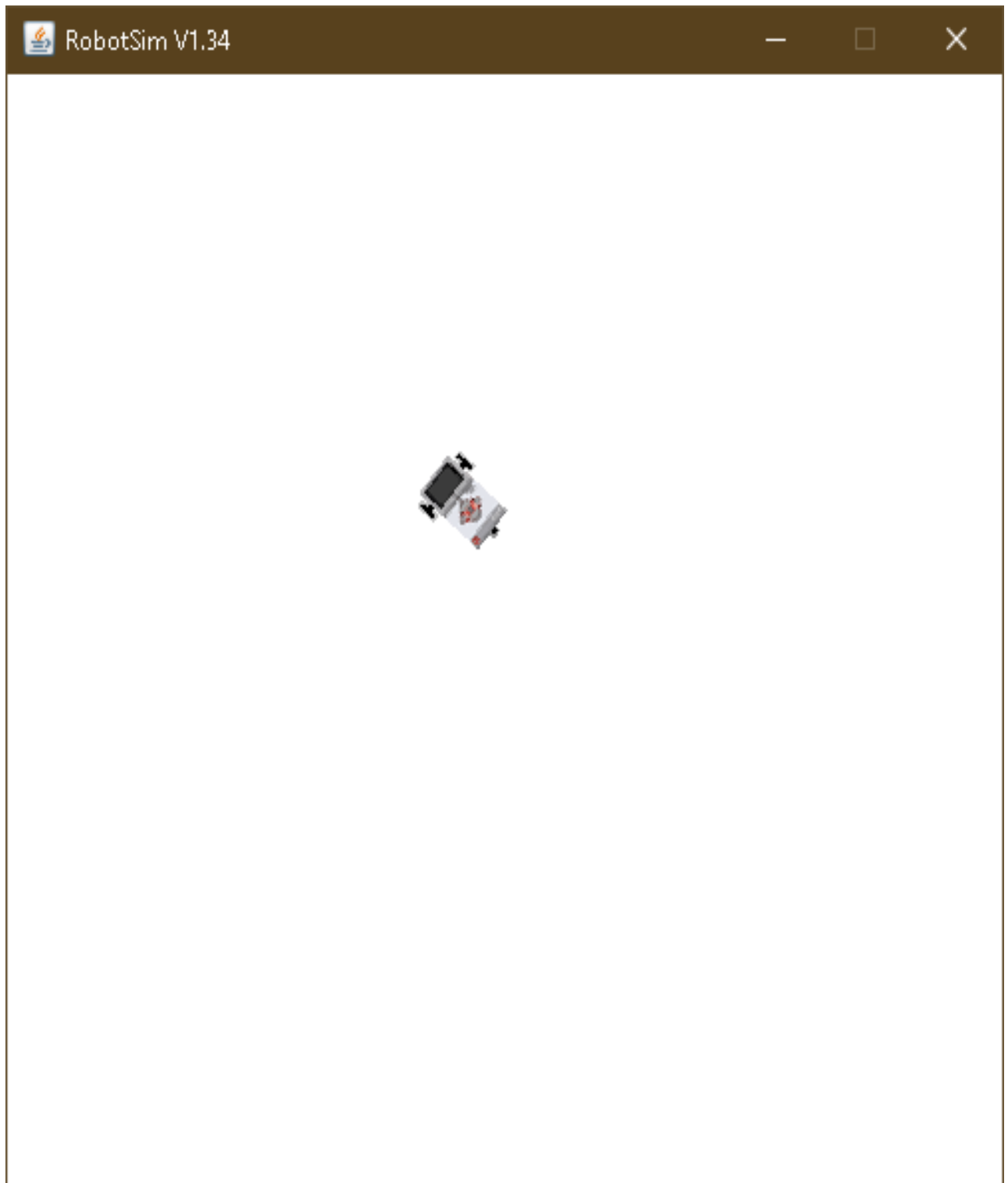
Aim: Write a program to create robot to perform a circular motion without using gear.

Code:

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

public class
Assignment_2b {
    Assignment_2b () {
        TurtleRobot t = new TurtleRobot
        (); t.setTurtleSpeed (100);
        while (true) {
            t.forward
            (2);
            t.left (2);
        }
    }
    public static void main (String []
        args) { new Assignment_1b ();
    }
}
```


Output:



Assignment 3:

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 ();
}
}
```

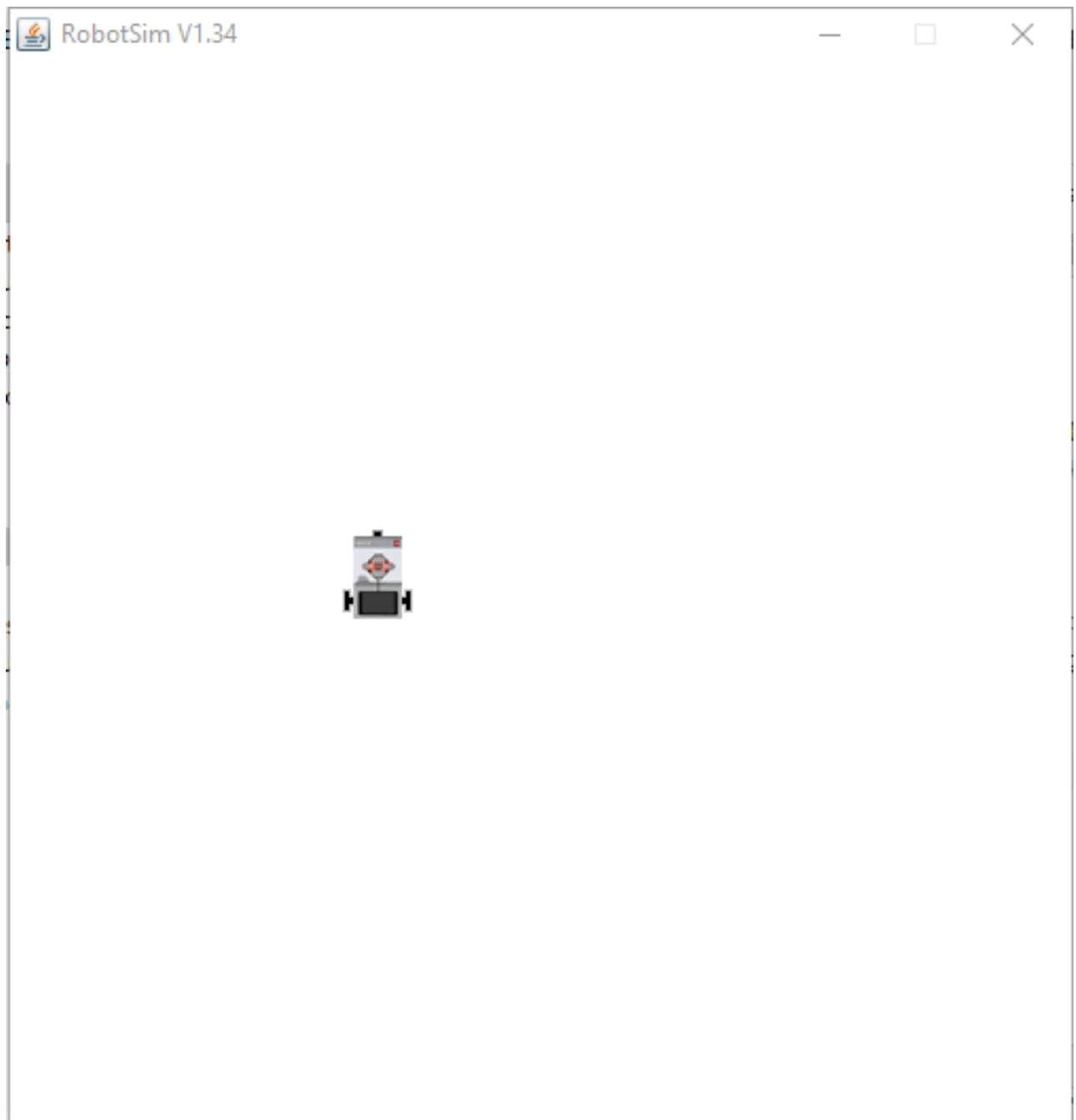
Output:

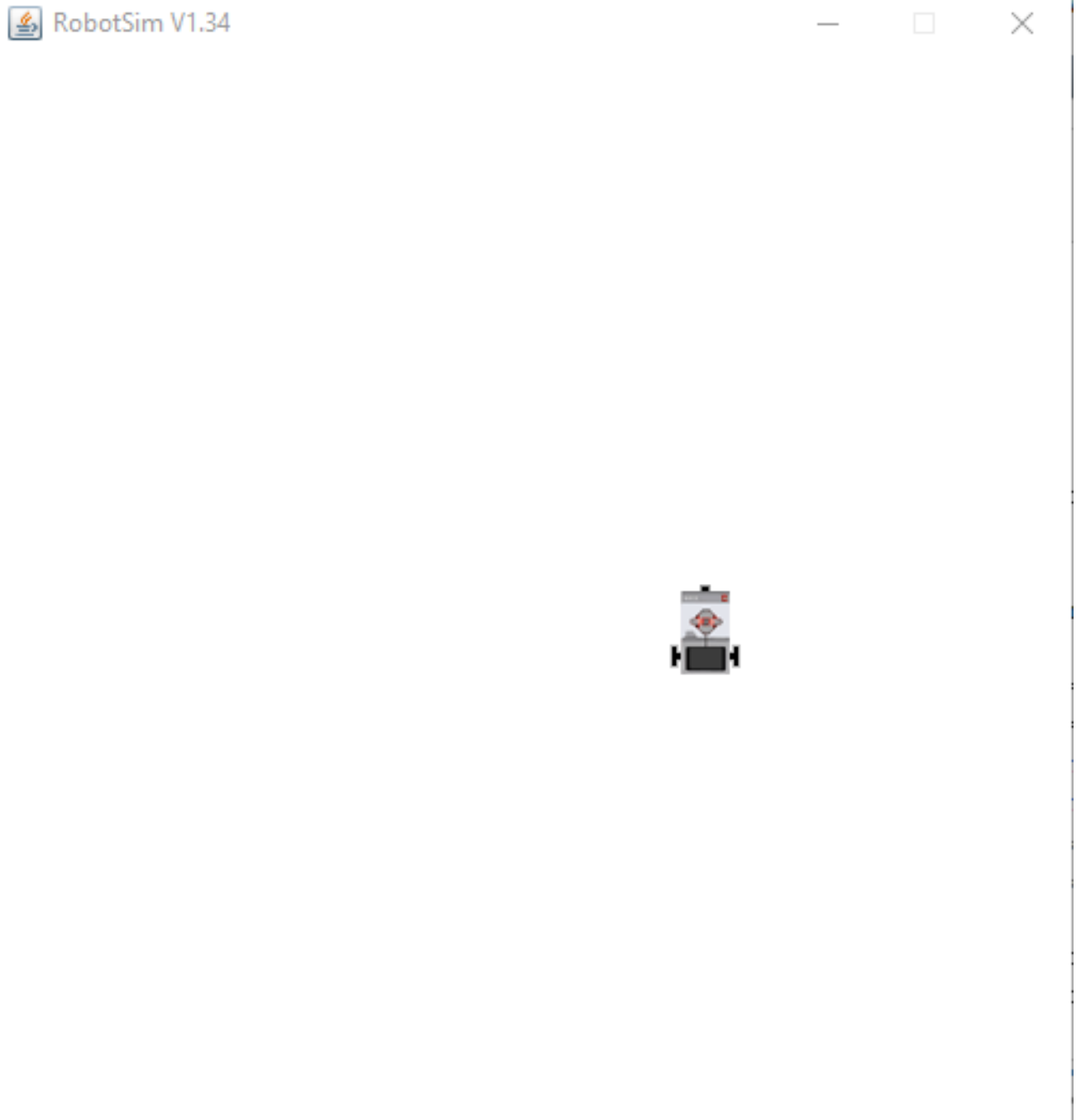


The screenshot shows a code editor window titled "Output - RoboticsAssignment (run)". The output text is as follows:

```
run:
Enter 1 for left and 2 for right:
1
|
```

At the bottom of the window, there is a status bar with the text "RoboticsAssignment (run)" and "running...". To the right of the status bar, there is a tab icon with the number "2", a timer showing "59:1", and a keyboard shortcut "INS".





```
Output - RoboticsAssignment (run) ×
run:
Enter 1 for left and 2 for right:
2
BUILD SUCCESSFUL (total time: 1 minute 24 seconds)
|
```

Activate Windows
Go to Settings to activate Windows.