

OrbBasic Lesson 2

Circles and if/then/else: Student Guide

In this lesson, you're going to create a new program with OrbBasic that makes the Sphero go in a circle. You'll be using variables again, and you'll also learn about the if/then statement, where Sphero will do something, but only if a statement is true.

Here are the Sphero commands you'll be using for this lesson:

- goroll – Makes Sphero roll at a given speed and heading. Also makes it stop.
- delay – Makes Sphero wait an amount of time before doing the next command
- goto – Makes Sphero go to a certain place in the program
- variables – Used to store a number
- if/then – Used to make the Sphero do something if a statement is true
- if/then/else – Like if/then, but also does something else if the statement is false
- LEDC – Makes the Sphero light up a color

First you have to connect Sphero to the iPad (Part 1), then you'll aim Sphero (Part 2), then you'll write an OrbBasic program to make Sphero roll in a circle (Part 3). Next, you'll make it light up with two different colors (Part 4), and then you'll control the distance it rolls with a variable (Part 4). Finally, you'll have a challenge to use the variable to make Sphero roll in bigger and bigger circles (Part 5).

How can we make the Sphero roll in a circle?

There's no "roll in a circle" command in OrbBasic. Instead, we have to create the circle out of small line segments. We'll start with a goroll command to roll the Sphero forward just a little bit at heading 0 degrees. Then we'll do another short goroll, but this time increase the heading just a tiny bit to 5 degrees. You can see in the diagram below that the top arrow is just slightly bent to the right of the bottom arrow, which is totally straight up and down.

Next, we increase it to 10 degrees and roll a little more. Then 15 degrees. Then 20 degrees. We add 5 degrees to the heading each time.

Once the heading goes all the way to 360 degrees, the Sphero will have moved in a full circle.



If/then and if/then/else statements

Sometimes you'll want to make Sphero do something, but only when certain conditions happen. For this, Sphero has an if/then statement. The if/then statement looks like this:

Line number
10 if x > 100 then goroll 0, 0, 0
Check if true
Only do this if true

This line is saying that if the variable x is greater than 100, then Sphero should stop. (A goroll command with all zeros will stop the Sphero.) The part after if usually involves a variable. The part after then can be anything you want Sphero to do.

You can also add an else to an if/then statement. This will allow you to tell Sphero to do something if a statement is true, but to do something else if it is not true. The if/then/else statement looks like this:

Line number

Check if true

Only do this if false

```
20 if b = 0 then delay 1000 else delay 2000
```

Only do this if true

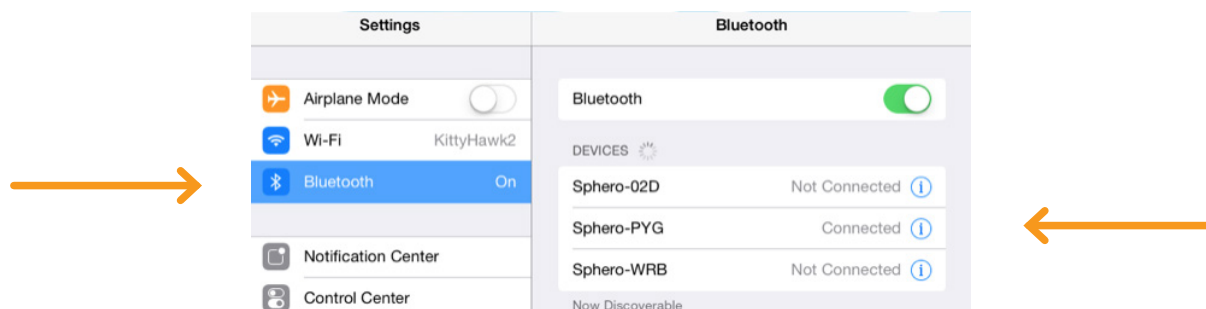
This line is saying that if the variable b is 0, then the Sphero should delay for 1000 milliseconds (1 second), but if b is not 0, then it should delay for 2000 milliseconds (2 seconds).

Let's try these out!

Part 1: Connect the Sphero

First thing you need to do is to connect the iPad to Sphero. Here's how:

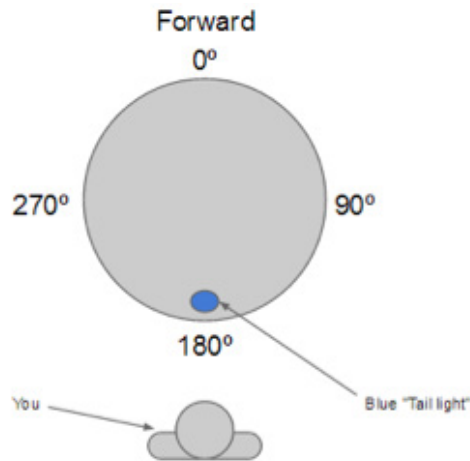
1. Pick up Sphero from its charging station and tap it twice on the logo to wake it up. You may have to tap it hard. It will start flashing colors when it is awakened out of its "sleep" state.
2. On your device, make sure Bluetooth is enabled. From the home page, click on Settings at the bottom. Then choose Bluetooth.
3. You will be shown a list of Spheros. Connect to the appropriate Sphero by tapping it. You can tell which Sphero is which by the names, which relate to the colors the ball is flashing. For example, if it flashes purple, then yellow, then green, then that is ball PYG. Select the one you want. Once successfully connected, it will say "Connected".



Part 2: Aiming Sphero

Sphero has a direction built into it that it thinks of as "straight ahead". This is called the orientation. The first thing we want to do is to aim the Sphero so that the orientation is on the path we want it to go. Each Sphero has a blue light inside of it called the "taillight", which is always on the exact opposite side of the straight ahead direction. You are going to set the taillight so that it's pointing right at you when you look down the path you want Sphero to go. Then, when it goes straight ahead, it will be on that path.

Part 2: Aiming Sphero (continued)

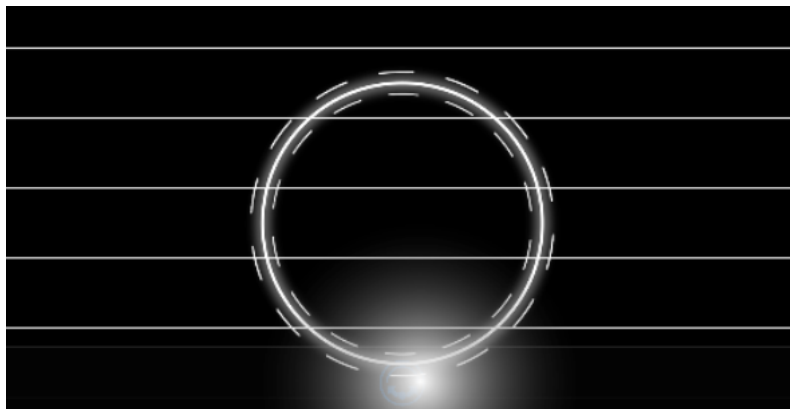


Follow these steps to aim the Sphero:

1. Go to the home screen and open OrbBasic.
2. Have one of you hold the Sphero and stand at the beginning of the path you will use for your experiments.
3. Now, you will aim the Sphero in that direction. Have a second member of the group use the iPad. In OrbBasic, you will see a circle with two arrows at the bottom center of the screen. Tap on it and hold it.



4. A white circle will appear. Move your finger slightly to rotate the insides of the Sphero. You will see a blue light inside the ball. Move it around until the blue light is directly facing the person holding the Sphero. This is the "taillight", and shows the direction opposite where the Sphero will move when moving straight ahead.



Important: For these experiments, the Sphero will travel a long distance, so be sure to aim the Sphero as accurately as you can to keep it on track. You can also re-aim Sphero anytime!

Part 3: Making Sphero roll in a circle

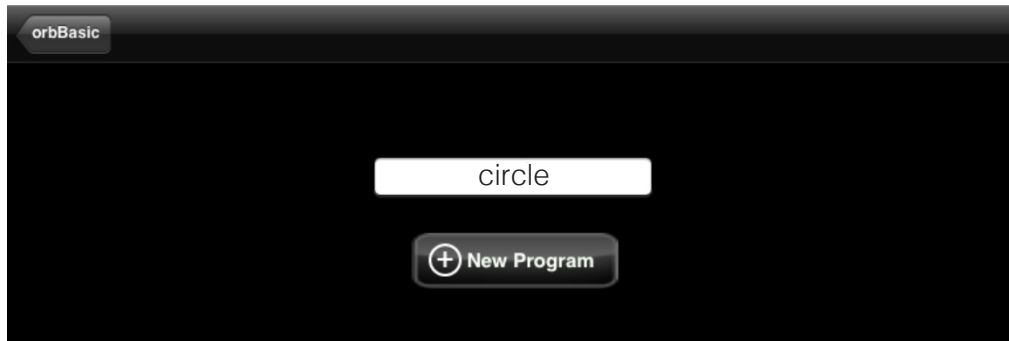
Now that we have the Sphero going in the right direction, let's get it going in a circle. As explained above, we want to increase the heading by 5 degrees each time. We'll need a variable called h (for heading) that starts at 0, and then we'll add 5 to it after each small roll. Follow these steps:

Part 3: Making Sphero roll in a circle (continued)

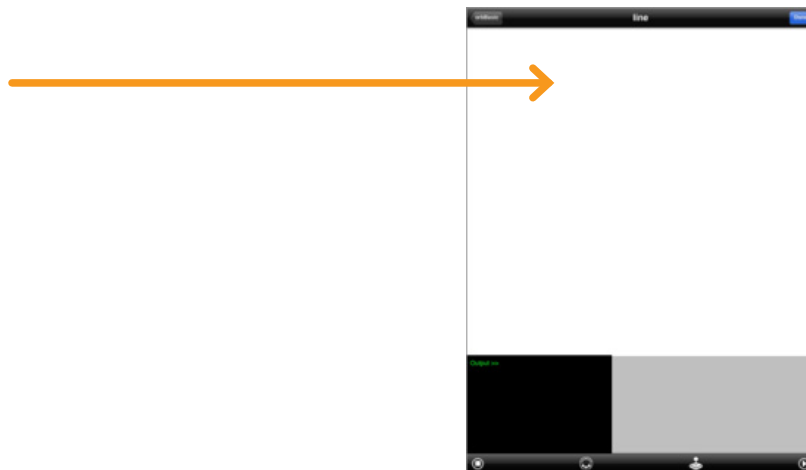
1. Tap the + button at the bottom to create a new program



2. In the space where it says Program Name, give your program a name, then click the '+ New Program' button under it.



3. Tap in the big white space. A keyboard will appear at the bottom of the screen.



4. Start typing code. You'll first need to set the variable h to zero.

```
10 h=0
```

5. Next, you'll want to roll at that heading (variable h) at speed 50, type 2 (turns quickly).

```
20 goroll h,50,2
```

6. Then, you'll want a very short delay so that it only rolls a little bit. Use 50 milliseconds.

```
30 delay 50
```

7. The next line will increase the heading variable h by 5 degrees.

```
40 h=h+5
```

8. The next line will increase the heading variable h by 5 degrees.

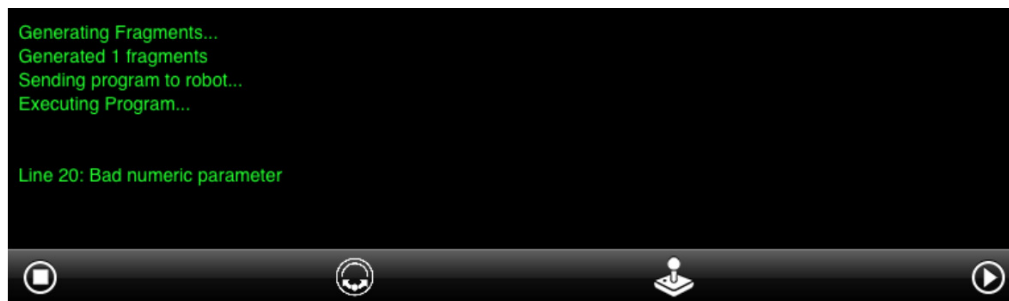
```
50 goto 20
```

Your full program should look like this:

```
10 h=0
20 goroll h,50,2
30 delay 50
40 h=h+5
50 goto 20
```

9. Tap the Done button and then the Play button to see what happens. But be ready to tap the Stop button.

Did it go in a circle? What happened when it came back to where it started? You probably had to tap the Stop button. Clearly something went wrong, and if you look in the black area, you'll see an error:



What could be going wrong at line 20? Well, h started at zero, and then went to 5, 10, 15, etc. When it got back to the starting point, h had a value of 360. The first number in the goroll command can only be between 0 and 359. So Sphero stopped the program and just kept doing the last command, which was to roll forward.

How can we fix this? We need to check if h is 360, and if it is, set it back to 0 so that it can keep going in circles. That's where the if/then statement comes in.

10. Add this line between 40 and 50:

```
45 if h=360 then h=0
```

11. Tap the Done button and then the Play button to see what happens. Is Sphero going in circles?

Part 4: Adding Color

Let's make your program a little fancier. It needs some color. OrbBasic has an RGB command, similar to what MacroLab has, but it also has an LEDC command that lets you choose a color from a table. (The type of lights that Sphero uses are called LEDs, for light emitting diodes. That's why the command has that name.)

Part 4: Adding Color (continued)

Here are the colors that you can use with LEDC:

For example, a command to light the Sphero orange would be:

```
45 if h=360 then h=0
```

Let's add a line to our program that makes the Sphero one color when the heading is 0 to 180 degrees (the first half of the circle) and another color when it's 180 to 360 degrees (the second half of the circle). This is a great place to use the if/then/else command.

1. Add a line in between lines 45 and 50 to check if the heading is less than 180 degrees. If it is, then light up one color. If it's not, light up another color. You get to choose the colors. (Use the table to figure out what numbers to use.) In the example below, it will be red for the first half, and green for the second half.

Important: OrbBasic is a case-sensitive programming language. This means that whether a letter is capital or not is important. LEDC must be made with all capital letters.

```
47 if h<180 then LEDC 1 else LEDC 2
```

2. Tap the Done button and then the Play button to see what happens. Tap the Stop button when you've seen enough.

Number	Color
0	No light
1	Red
2	Green
3	Blue
4	Orange
5	Purple
6	White
7	Yellow

Part 5: Challenge

Let's take what we learned in OrbBasic lesson 1 and apply it here. Remember how we had the Sphero going farther and farther each time? Now you'll modify your program so that it makes bigger and bigger circles. Each time it makes a circle; it should increase the size of the circle by a little bit.

Here are some hints:

1. The way to make the circle bigger is to make the delay bigger.
2. You'll need a variable to use for the delay. Take a look at your OrbBasic 1 program to see how you did this before.
3. You'll only want to increase the delay when the Sphero has gone around a full circle. That means the line to increase it should be an if/then statement, where you are checking to see if the heading (the variable h) is at zero.
4. You'll only want to increase the delay a little bit each time. Try starting at 50 and then increasing by 10 each time it has gone full circle.

Good luck and have fun!