



OrbBasic 1: Student Guide

Sphero MacroLab is a really cool app to give the Sphero commands, but it's limited in what it can do. You give it a list of commands and it starts at the top and goes to the bottom, but it can't do anything fancier than that. To make programs that can do more, you need an app called OrbBasic. OrbBasic uses a text-based programming language and really increases what you can do with a Sphero.

You are going to be using Sphero to learn about two important concepts in programming: the goto ("go to") statement, and variables.

- 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

First you have to connect the Sphero to the iPad (Part 1), then aim your Sphero in the correct direction (Part 2), then you'll write your first OrbBasic program to make Sphero roll out and back (Part 3), then you'll make it repeat itself (Part 4), and then you'll control the distance it rolls with a variable (Part 5). Finally, you'll have a challenge to use the variable to make the Sphero roll less and less (Part 6).

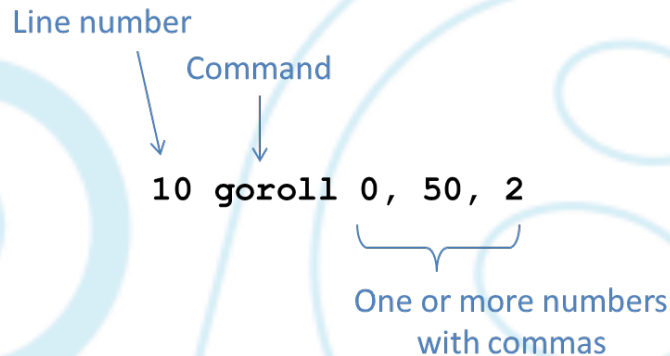
What is OrbBasic?

OrbBasic is a programming language. A programming language is a list of instructions that tells a computer what to do. Although MacroLab allowed you to create programs, it was done using a graphic interface, meaning that you tapped buttons on the screen, slid sliders, and filled in boxes. Most programming languages use text instead, and this is what OrbBasic does.

Why do we call it a language? Usually a language is used for two people to communicate something. For a programming language, it's a person communicating with a computer. One of the big differences between a programming language and a human language is that when you

communicate to a computer, you have to get the words exactly right. If you make a small mistake, either the computer won't understand it, or it won't do what you want it to do.

Most commands in OrbBasic looks like this:



The first number is the line number. These are needed for goto ("go to") commands, telling the computer to jump to that line number. Usually lines are number 10, 20, 30, etc. You could number them 1, 2, 3, etc., but if you want to put a new line in between the first and second one, if you go up by 10s, then you can use 15 as a line number between 10 and 20. If you go up by 1s, then there's no space to put any lines in between 1 and 2.

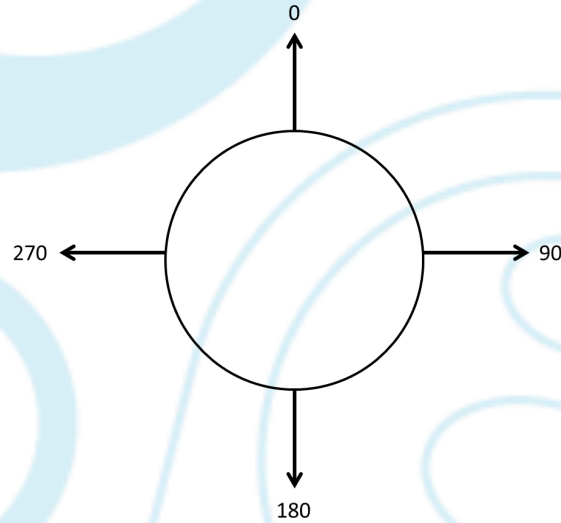
The second piece of text is the command. This will tell Sphero to do a particular action.

The numbers at the end are information that is used by the command. Sometimes there is one number, and sometimes there are many. If there are many, then you put a comma in between each of them.

Making the Sphero roll and stop

To make Sphero roll or stop, you use the `goroll` command followed by three numbers:

- First number: heading. From 0 to 359, it is the number of degrees from straight ahead (see diagram below)
- Second number: speed. In MacroLab, the speed value went from 0% to 100%. In OrbBasic it goes from 0 to 255. (Why 255? It has to do with how computers store numbers.)
- Third number: type of rolling. It can be 0 (to stop), 1 (normal), or 2 (fast rotation)



So, for example, this command:

```
30 goroll 90, 128, 2
```

means roll at heading 90 degrees, at a speed of 128 (about 50%), expecting to change direction quickly. 30 is the line number, which means it is most likely the third command in the program.

Notice that unlike MacroLab, there is no delay. Sphero will keep rolling at that heading and speed until you tell it to do something else. So we typically add a delay after that step. The delay command has one number after it, which is the number of msec (1/1000 of a second) to wait for. For example, this command:

```
40 delay 5000
```

means wait for 5000 milliseconds (5 seconds) to do the next command. The line number of 40 means that it is probably the line after the goroll line.

Finally, when you want the Sphero to stop, you should use the goroll command with zeroes for speed and type. For example,

```
50 goroll 0, 0, 0
```

means point in the heading direction of 0 degrees and stop.

Jumping to a new place in the program

The goto command is used to jump to a new place in the program. (goto is a combination of the words "go" and "to".) It has one number after it, which is the line number to jump to. For example, this code:

```
60 goto 20
```

means to go to line 20 and perform the command at that line. Once Sphero is done with that command, it should do the next one.

For example, this code will roll and stop the Sphero over and over, until you stop the program.

```
10 goroll 0, 50, 2  
20 delay 1000  
30 goroll 0, 0, 0  
40 delay 1000  
50 goto 10
```

Variables

You will also learn about variables in this lesson. Variables are a place in memory where a computer stores a number. Every variable has a name, and in OrbBasic, the variable names are one letter long, and can be anything in the alphabet except z. (z is used for something else.)

For example, this command:

```
50 b = 35
```

means that you are storing the number 35 in the variable named b. Then, you can use the variable just like you would a number. For example,

```
60 goroll b, 128, 2
```

means that you are rolling the Sphero with a heading of 35 degrees.

The cool thing about variables is that they can change what number they hold. For example, this command:

```
70 b = b + 5
```

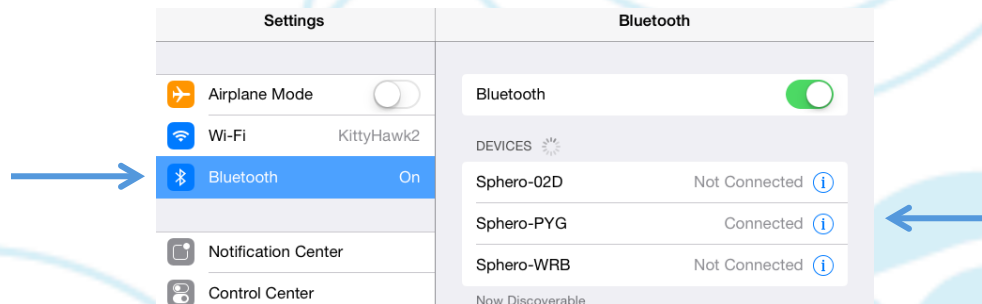
means to take what's stored in the b variable, add 5, and then store the new number in the b variable. If b were 35, then after doing this line, it would be 40.

Don't worry if this doesn't make sense just yet. You'll learn more when you do the exercises. Let's get started!

Part 1: Connect Sphero

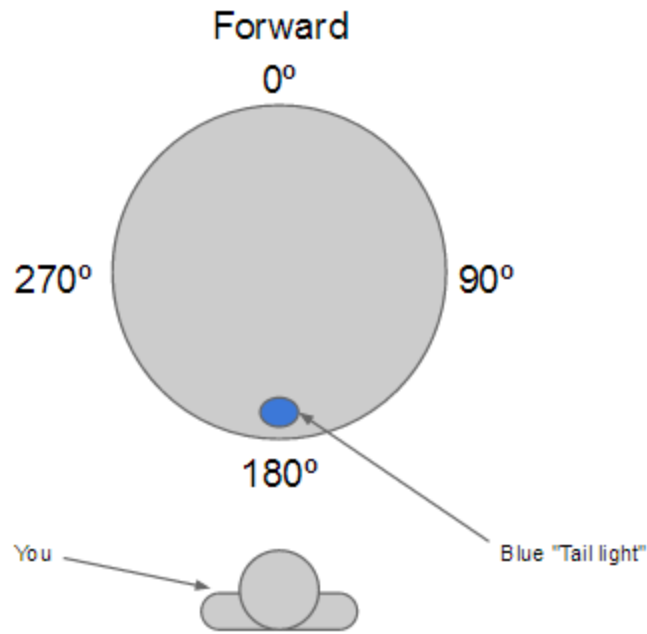
First thing you need to do is to connect the iPad to Sphero. Here's how:

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.

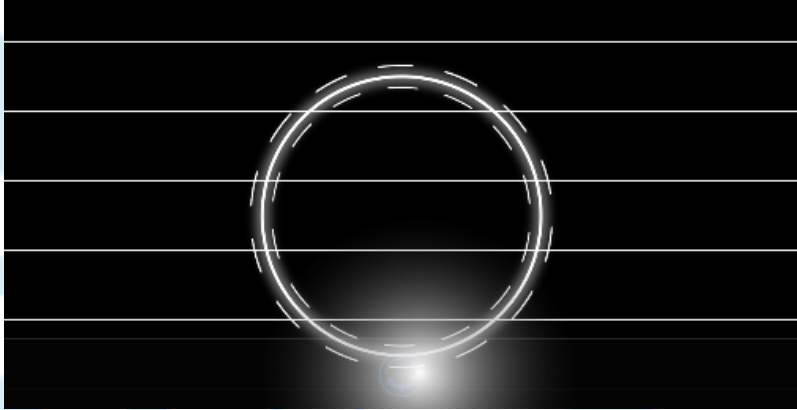


Follow these steps to aim the Sphero:

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.

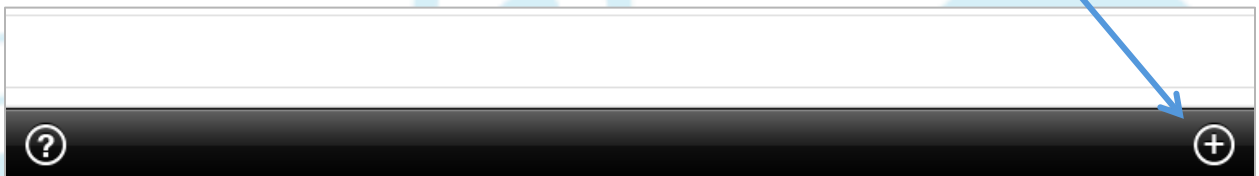


Important: For these experiments, the Sphero will travel a long distance, so be sure to aim the Sphero as accurately as you can to keep it on track. You can also re-aim Sphero anytime!

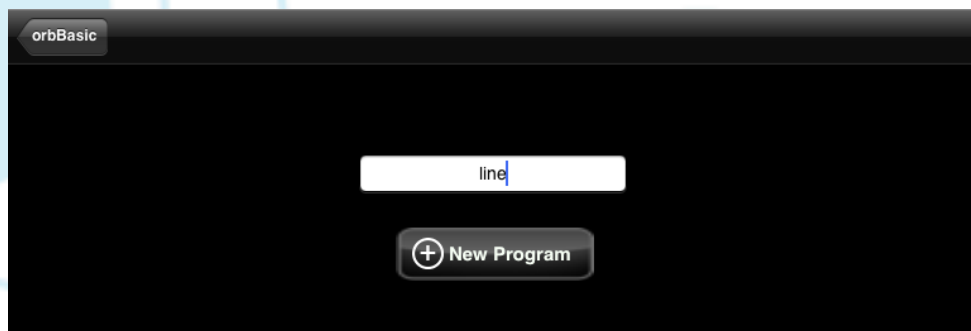
Part 3: Your first OrbBasic program

Now that we have Sphero going in the right direction, let's write your first OrbBasic program. It's going to simply roll a distance and then roll back and stop.

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. Type this code to roll Sphero forward at speed 50 for 2 seconds, then roll it back (180 degrees heading) at speed 50 for 2 seconds, and then stop. We'll use a roll type of 2 (last number), so that it can switch directions quickly.

```
10 goroll 0,50,2  
20 delay 2000  
30 goroll 180,50,2  
40 delay 2000  
50 goroll 0,0,0
```

5. Tap the Done button in the upper right hand corner.



- Now, you need to aim Sphero. Use the aiming button at the bottom of the screen, just like you did with MacroLab.



- Tap the Play button to see the Sphero move back and forth.



Congratulations! You wrote your first OrbBasic program!

Part 4: Goto

So far, that program would have been easier to write in MacroLab. But now let's do something that we couldn't do in MacroLab. Let's make Sphero roll back and forth, over and over. To do this, add a goto command at the end to jump back to the beginning. Change line 50 to be:

```
50 goto 10
```

Your new code should look like this:

```
10 goroll 0,50,2
20 delay 2000
30 goroll 180,50,2
40 delay 2000
50 goto 10
```

- Now we should see the ball rolling back and forth. Tap the Play button to see it work.
- Tap the Stop button when you've seen it roll back and forth enough times.



Part 5: Variables

Variables are a powerful way to make Sphero do things. As mentioned in the introduction, variables store numbers in Sphero's memory. Let's use a variable called d (for delay), and have

that store the delay value. We need to set it up before we start rolling, so we will take advantage of the fact that we start at line 10, and add a new line 5 at the top.

1. Add this line at the very beginning:

```
5 d=2000
```

Warning: When adding a line at the top, it can be easy to accidentally tap orbBasic and end up back at your list of programs. If that happens, just tap on line to get back to your program and very carefully tap at the top of the program. Once you are at the top, tap Return to add a new line.

2. We have now stored the number 2000 in the variable d. Let's use it by replacing the 2000s in the delay line with d. Your code should look like this now:

```
5 d=2000
10 goroll 0,50,2
20 delay d
30 goroll 180,50,2
40 delay d
50 goto 10
```

3. Tap on Done and Play. The Sphero will roll the same way as before.

So what's the use of that? Well, having something stored in memory can be very powerful. Let's say we wanted to change it to roll 3 seconds instead of 2. Instead of changing all the 2000s to 3000s in the delay commands, we just need to change our first line to:

```
5 d=3000
```

4. Change that first line from 2000 to 3000. Run the program, and you'll see that the Sphero goes farther now.

So that was convenient, but not very exciting. Let's really make use of variables by adding a line that adds half a second to the delay time. Then each time, it will roll a little longer and a little farther.

5. First, set the value back to 2000 in the first step (step 5). Then add a new line between 40 and 50, that will add 500 milliseconds (half a second) to d each time. Using a step number of 45 lets us put it between 40 and 50.

```
45 d=d+500
```

Your code should look like this now:

```
5 d=2000
10 goroll 0,50,2
20 delay d
30 goroll 180,50,2
40 delay d
45 d=d+500
50 goto 10
```

6. Tap the Play button. The Sphero will roll back and forth, slightly farther each time.

Part 6: Challenge

Can you figure out how you would change the program so that it starts off rolling for 5 seconds out and back, and then rolls one second less each time? Remember that the delay is in milliseconds (1/1000 of a second), so see if you can figure out what the number should be.

Once the delay gets down below zero, the program will stop working properly, and the Sphero will just start rolling in a straight line. Tap Stop at that point.

Once you know you have it working, write down your program on your worksheet. Have Fun!