



OrbBasic Lesson 2

Circles and if/then/else: Teacher Guide

Overview

Students will use Sphero to explore the computer science concepts of variables and conditionals (if statements). They will use OrbBasic, which is a text-based programming language for the Sphero. They will write a simple program that rolls Sphero in a circle until it gets an error. Then they will fix the error by adding an if/then statement. They will learn about if/then/else statements to light up Sphero under certain conditions. For the challenge, they will make the circle increase its radius each time it goes around a circle.

Read through the student guide to learn about what the if/then, if/then/else, and LEDC commands do. At the start of the lesson, discuss with the students how to program Sphero to roll in a circle, and how if/then and if/then/else statements works.

Objective

Students will:

- Create an OrbBasic program to roll Sphero in a circle once, using a variable to store the heading
- Modify the program with an if/then statement to fix an error and make it go in a circle indefinitely
- Modify the program with an if/then/else statement to light up one color half the circle, and another color the other half
- Modify the program to increase the size of the circle at the end of each cycle.

Common Core Math Standards

The following Common Core Math Standards for 4th and 5th grade apply to this lesson:

- [CCSS.MATH.CONTENT.4.OA.C.5](#): Generate and analyze patterns

- [CCSS.MATH.CONTENT.5.OA.B.3](#): Analyze patterns and relationships
- [CCSS.MATH.PRACTICE.MP1](#): Make sense of problems and persevere in solving them.
- [CCSS.MATH.PRACTICE.MP2](#): Reason abstractly and quantitatively.
- [CCSS.MATH.PRACTICE.MP4](#): Model with mathematics.
- [CCSS.MATH.PRACTICE.MP8](#): Look for and express regularity in repeated reasoning.

Materials Needed

Spheros are controlled via Bluetooth on either Apple (iPod, iPhone, or iPad) or Android devices. Ideally, you would do this lesson in groups of 3 or 4 students, each with their own Sphero and device. This lesson is designed for iPads, but other devices could be used. Here is what each group would need:

- iPad with Sphero OrbBasic loaded. You can get Sphero OrbBasic for free from the iTunes app store.
- Sphero that has been fully charged
- Print-out of the worksheet (last page of teacher's guide)
- A flat clear area of at least 10 x 10 feet. (Preferably not very slippery.)

Although not required, it is very helpful to have a keyboard attached to the iPad.

Part 1: Connect the Sphero

In part 1, students need to connect each iPad with a Sphero. They will:

1. Wake up the Sphero
2. Turn on Bluetooth
3. Connect the correct Sphero to the iPad, using the colors that it flashes as a way to tell which Sphero has which name

Part 2: Aim the Sphero

In part 2, students need to set the orientation, which is the direction of 0 degrees heading for Sphero. This is called "aiming". It's important that they get this right so that the Sphero will follow the path and not bump into anything. To do this, they need to adjust the blue "taillight" so that it is pointing directly at them. If they do this correctly, then the Sphero will roll directly away from them. Students will:

1. Open up OrbBasic on the iPad
2. Hold the Sphero in front of them as they look down the path

3. Tap and hold the aim icon at the bottom of the screen and adjust the taillight so that it is pointing directly at them.

Part 3: Making the Sphero roll in a circle

In part 3, students will create an OrbBasic program that rolls Sphero in a circle by having it roll a very short distance and then increase the heading by 5 degrees. It uses a variable called h that starts at zero and is increased each time by 5. See the student guide for the code.

The first version of the code lets the variable h start at 0 and keep getting larger. When it hits 360, then the roll command gets an error because heading values need to be between 0 and 359. You can see the error at the bottom part of the screen. At this point, the Sphero stops running the program and keeps doing whatever the last step was, which is to roll. So the program makes the Sphero roll in a circle and then keeps rolling in a straight line. They will need to tap Stop to make the Sphero stop.

They will fix the error by adding a new line with an if/then statement which checks to see if h has a value of 360, and if it does, it sets it back to 0. With this line, Sphero will roll in circles indefinitely.

Part 4: Adding Color

Students will learn about the if/then/else statement, which will do one command if something is true, and a different one if it is not. In this case, it checks to see if $h < 180$ (the heading is in the first half of the circle), and if it does, it lights it up in one color; if it is not, then it lights it up in another.

Although you could use OrbBasic's RBG command for the colors, OrbBasic has a simpler command called LEDC. (LED refers to the light emitting diodes that are used to light up the Sphero.) LEDC is followed by one number, which corresponds to a color. The student guide has a table that shows which color has which number.

Part 5: Challenge

For the challenge, students will see if they can modify the program to increase the size of the circle each time the Sphero goes around. This is a fairly difficult challenge, so several hints are given. If they look back at their previous lesson's program, they should be able to figure out that they need a variable (for example, d) that holds the delay. They need to add a line to start it at 50, then modify the delay commands to use it, and then add an if/then statement to

increase by 10, but only if $h=0$. The answer is below (the line numbers and LEDC values don't have to be exactly the same):

```
10 h=0
15 d=0
20 goroll h,50,2
30 delay d
40 h=h+5
45 if h=360 then h=0
47 if h > 180 then LEDC 1 else LEDC 2
48 if h=0 then d=d+10
50 goto 20
```