OrbBasic Lesson 1
Goto and Variables: Teacher Guide

Overview

Students will use Sphero to explore the computer science concepts of program flow and variables. They will use OrbBasic, which is a text-based programming language for the Sphero. They will write a simple program that rolls Sphero out a distance and back. Then they will modify it by making it repeat until stopped, and then they will use a variable to modify the amount of time the Sphero is rolling, eventually making the variable increase over time. For the challenge, they will make the variable decrease so that the Sphero rolls shorter and shorter distances.

Read through the student guide to learn about how OrbBasic programs are structured, and what the goroll, delay, and goto commands do, as well as what variables are. At the start of the lesson, discuss these concepts with the students.

Objective

Students will:

- Create a short OrbBasic program that rolls Sphero out a distance and back, and then stops.
- Modify the OrbBasic program to add a goto statement that goes back to the beginning
- Modify the OrbBasic program to add a variable that holds the delay time
- Modify the variable value and show that it affects the delay time everywhere it's used
- Add a line to increase the delay variable after each time out and back.
- Modify the program to decrease the delay variable each time out and back.

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
- A flat clear path of at least 25 feet. (Preferably not very slippery.)

Although not required, it can be 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: Your first OrbBasic program

In Part 3, students will create an OrbBasic program that rolls the Sphero out and back. See the student guide for the code. A few notes about OrbBasic:

- OrbBasic goroll commands do not have a delay. This means that you need to have first a goroll command followed by a delay command.
- OrbBasic roll speeds range from 0 to 255. (255 is one less than 2 to the power of 8, and since computers store numbers in powers of 2, 255 is a more convenient number from the computer's point of view.

Note: The code has to be exactly right for the computer to understand it. Look for error messages in the black space below the code to see if something is wrong. For example, in this case, the program has a heading of 1800 instead of 180 on line 30.

```
30 goroll 1800,50,2
```

You can see the error message below, and how it mentions which line number the error occurred on.

Part 4: Goto

Part 4 involves replacing the last line of code that made the Sphero stop with a code that jumps back to the beginning. This will result in the Sphero executing the program indefinitely. Again, see the student guide for the code. Have them tap the Stop button when they have seen enough.

Part 5: Variables

Variables may be a difficult concept for students. The idea is simply that there is a space in Spheros memory to hold a number. In this section, we use that variable to hold the amount of delay time. Remember, the delay time is how long the Sphero will be rolling for, so if we increase it, the Sphero will go farther. In the first part of this section, students add a new variable called d and set it to 2000. Then they replace the delay values with d. When they run the code, it does the same thing as before. The reason that having a variable is useful here is that you can now change the delay time in one place (the first line), and it will change everywhere it is used. So they can change it to 3000 in one place, and the Sphero will now roll for 3 seconds instead of 2.

In addition, they can add code to modify the delay time while the program is running. They'll add a line to add half a second (add 500) to d, and then each time the Sphero rolls away and back, it will do it for half a second more. This means it will go out longer and longer each time.

Note that to squeeze in new lines of code between existing ones, they will use line numbers like 5 and 45 instead of even multiples of 10 like they started with. This is why we start with lines spaced by 10, so it's easy to squeeze new ones in.

See the student guide for the code.

Part 6: Challenge

For the challenge, students will see if they can modify the program to start the delay at 5 seconds and make it drop by 1 second each cycle. The answer is below:

```
5 d=5000
10 goroll 0,50,2
20 delay d
30 goroll 180,50,2
40 delay d
45 d=d-1000
50 goto 10
```