Hello there, and thanks for taking a look at Sphero and Education!

The lessons in the SPRK program teach math, physics, and computer science concepts using hands-on, engaging activities with Sphero, a robot ball. Students work in small groups to write computer programs that control how the Sphero rolls and appears. They are designed as lessons primarily for 4th and 5th graders that will take approximately one hour.

These lessons start with an introduction and then list the Common Core Math Standards that are relevant to the lessons. They contain a teacher guide, a worksheet for the students to fill out, and a student guide.

The CORE lessons cover:
• Math: Percentages, division, geometry, and patterns
• Physics: Speed, time, and distance
• Computer Science: Program flow, variables, conditionals, and reading sensors

What is Sphero?

Sphero is a robot ball with several features that can be controlled through mobile apps, including computer programs that the students build. The main features are:
• Rolling. The Sphero can roll at a given speed and heading for a given amount of time.
• Colors. The Sphero can light up in any color.
• Bluetooth. Sphero connects to devices such as iPads, iPhones, and Android phones and tablets through wireless Bluetooth connections. This allows the Sphero to be controlled by a number of apps.

There are 4 education related apps available to control Sphero. Each of these is available for free from app stores such as iTunes and Google Play.

• Sphero. This is the main Sphero app used for firmware updates and general driving.
• Draw and Drive. Allows you to draw a shape with multiple colors and have Sphero roll in that shape and color.
• MacroLab. Creates simple programs ("macros") that are a series of instructions for the Sphero through an easy-to-use graphical user interface.
• OrbBasic. Creates more complex programs using a text-based programming language.

The CORE & STEM lessons in the SPRK program use MacroLab and OrbBasic.

Help and support

We're here for you! If you have questions, comments, suggestions, or just want to chat please contact us!

• General Support Email: support@gosphero.com
• General Support Phone: 1 (303) 502-9466
• Education: education@orbotix.com
• Volume Purchase: vpp@orbotix.com
• Twitter: @SpheroEdu
Heading and Aiming

One of the things that makes Sphero so unique is that its heading is relative to the user, not relative to the ball. This makes the Sphero much easier to get to go where you want it to go. The diagram shows how the heading works. Note that only 90 degree increments are shown in the diagram, but you can specify the heading down to 1 degree.

Each time the Sphero is turned on, it needs to be “aimed,” which means setting the direction that will be used for a heading of 0 degrees. This is accomplished with Sphero’s “taillight”. The taillight is a blue, light inside the Sphero. Each Sphero app has a button that lets you set the taillight, which looks like this:

To use this button, tap and hold on it, and then slowly move your finger around the circle. You will see the blue taillight rotate. When it is pointing directly at you (in other words, directly away from the direction you want the Sphero to roll for heading of 0 degrees), remove your finger. The student guides for all of the MacroLab lessons lead you through how to do this.

For an interactive introduction on how to aiming, use the Sphero app.
Overview

Students will use Sphero to explore the computer science concepts of reading sensors and storing data in variables. They will use OrbBasic, which is a text-based programming language for the Sphero. They will write a simple program that detects when Sphero is in the air and also when Sphero collides with an object and then when these events happen, Sphero will change color or behavior.

In OrbBasic Lesson 2 students learned about conditional statements such as if/then/else. In this lesson, students will use these conditional statements in conjunction with the function that reads the accelerometer data in order to build a program that reacts to sensor readings. Students will also use knowledge from Lesson 2 about changing Sphero’s color using LEDC. At the conclusion of this lesson students will work with the random function to make Sphero move at a random heading after a collision.

Read through the student guide to learn more about the accelone and random functions. At the start of the lesson, discuss with students about acceleration and accelerometers.

Objective

Students will:
• Create a short OrbBasic program that changes the color of Sphero when it senses that it is in the air.
• Create another OrbBasic program that recognizes when Sphero collides with something and then changes color.
• Create a third OrbBasic program that sends Sphero in a random direction after a collision.

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:
Materials Needed (continued)

- 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 open space. (Preferably not very slippery.)
- Objects to have Sphero collide with
  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: Sensing when Sphero is in the Air

In part 3, students will create an OrbBasic program that changes color when Sphero’s accelerometer senses that Sphero is in the air. The new command students will use in this part is called accelone. Accelone is set to be 0 for a perfectly calibrated Sphero in freefall (if the only acceleration acting on it is gravity). Due to the fact that few things are perfect, students will need to build in a soft buffer to determine if Sphero is in fact in the air. This will be done by coding if accelone< 200 instead of if accelone=0.

Students will use a conditional statement and the LEDC commands to change the color of Sphero when it is in the air. If the condition is false then the program will loop back to the beginning. See the student guide for the code.

Part 4: Sensing Collisions

For Part 4 students will create a macro similar to the one they built in part 3 but it will be sensing collisions instead of whether or not Sphero is in the air. Students will be building a code to read if the accelerometer value is greater than 5000 (which indicates that Sphero has collided with something).
Part 4: Sensing Collisions (continued)

If this conditional statement is true the program will move down to the next line of the program and follow a set of commands that change Sphero's color to red for 1 second. However, if the conditional statement is false then the program will utilize a goto statement that goes back to the beginning of the program. After building the program students can use the joystick in OrbBasic to test the program, it should turn red when it collides with another object.

Part 5: Challenge

In part 5, students will see if they can build a code that makes Sphero drive straight and then move in a random direction after a collision. The heading range for Sphero is 0-359 degrees, instead of 1-360 degrees. To ensure the program doesn't receive an error, we have to make the highest possible value for random to generate 359. In this program students may want to use line 5 to set the initial heading to 0 for Sphero. Using lines be useful because we only want the heading to be set once at 0 as an initial condition and then the program could then loop through the commands starting at line 10 as in the other macros.

For the challenge, ensure you have some open space in your classroom with a few obstacles set up. When planning which objects Sphero will run into, try to avoid using things like walls because it will be harder to see what direction Sphero travels after the collision. The answer to the challenge is below (the line numbers and LEDC values don't have to be exactly the same):

```plaintext
5  h=0
10 LEDC 2
20 goroll h, 100, 1
30 if accelone> 5000 then goto 40 else goto 10
40 LEDC 1
50 heading rnd 359
60 delay 500
70 goto 10
```
Overview

In this lesson, you’re going to create a new program with OrbBasic that makes the Sphero change colors when it senses it is in the air. You’ll be using variables and conditional statements again, and you’ll also learn about Sphero’s accelerometer and the accelone and random functions.

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
- **Accelone** - returns the effective acceleration vector that Sphero is experiencing.
- **Heading** - the direction Sphero will move, range 0-359 degrees
- **Rnd** - returns a random number between 1 and the value provided

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 turn purple when it is in the air (Part 3). Next, you’ll make it light up with a different color when it collides with another object (Part 4). Finally, you’ll have a challenge to use the accelone and random functions to make Sphero roll in a random direction after sensing a collision (Part 5).

A Quick Review

Let’s go through a quick review of what you learned in the last lesson. The if/then statement looks like this:

```
10 if x > 100 then goroll 0, 0, 0
```

Only do this if true
A Quick Review (continued)

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.

Also as a review from OrbBasic lesson 2, these are the colors that you can use with LEDC:

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

```
50 LEDC 4
```

### 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”.

<table>
<thead>
<tr>
<th>Number</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No light</td>
</tr>
<tr>
<td>1</td>
<td>Red</td>
</tr>
<tr>
<td>2</td>
<td>Green</td>
</tr>
<tr>
<td>3</td>
<td>Blue</td>
</tr>
<tr>
<td>4</td>
<td>Orange</td>
</tr>
<tr>
<td>5</td>
<td>Purple</td>
</tr>
<tr>
<td>6</td>
<td>White</td>
</tr>
<tr>
<td>7</td>
<td>Yellow</td>
</tr>
</tbody>
</table>

### 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.

Part 3: Making Sphero turn purple in the air

Sphero has an accelerometer inside, an accelerometer is used to measure the acceleration, or change in velocity of Sphero.
Orb Basic has a command called accelone which is a cumulative reading of the accelerations acting on Sphero at a given time in all directions (X, Y, Z). If Sphero is only being acted on by gravity (i.e. when it is in the air) the reading will be zero. Due to the fact that few things are perfect, you will need to build in a soft buffer to determine if Sphero is in fact in the air. You can set up the buffer so that if the value is less than 200 that corresponds with Sphero being in the air. You can build a very short 2 line program to change Sphero's color when it is in the air.

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. In the first line of code for your new program you want to check whether the accelone value is sensing that Sphero is in the air or not. To do this a conditional statement is very useful. If the accelone value is less than 200 you will want Sphero to turn purple using LEDC 5 and if the accelone value is greater than 200 you will use an else statement to keep Sphero dark. To do this you can use the line below:

   `10 if accelone < 200 then LEDC 5 else LEDC 0`

5. Next, the program needs to go back to the beginning and run through the conditional statements again

   `20 goto 10`

The final code will look like:

```
10 if accelone < 200 then LEDC 5 else LEDC 0
20 goto 10
```

6. Tap the Done button and then the Play button and toss Sphero in the air to see what happens.

**Part 4: Collision Detection**

For Part 4 you will create a macro very similar to the one you just built but it will be sensing collisions instead of whether or not Sphero is in the air. When the accelerometer value is greater than 5000 that indicates that Sphero has collided with something.

1. First you can make the default color green

   `10 LEDC 2`
2. Using accelone, detect if a collision happens

   \[
   20 \text{ if } \text{accelone} > 5000 \text{ then goto 30 else goto 10}
   \]

3. Now you need to set Sphero to turn red using the command below for the case where the conditional statement is true

   \[
   30 \text{ LEDC 1}
   \]

4. In order to ensure it doesn’t just blink red but stays red for an entire second after it hits an object the next line should be a delay

   \[
   40 \text{ delay 1000}
   \]

5. Finally the program needs to jump back to beginning

   \[
   50 \text{ goto 10}
   \]

6. Tap the Done button and then the Play button. Use the joystick to drive Sphero

   Does your Sphero turn red when it runs into something?

**Part 5: Challenge**

Now that you have built a program that reacts after a collision, try to build a slightly more advanced program where Sphero drives in a straight line but if it collides with an object it changes color and rolls off in a random direction. This will combine topics from previous lessons as well. Here are some tips to guide you through building this program.

1. You will need to set up a variable h, which is the heading value for sphero and to begin you will need to set your initial heading to 0.

2. You may want to set an initial color for Sphero as well.

3. Next have Sphero to roll in a straight line

4. Program Sphero to change colors and move at a random direction if it senses a collision.
   (Tip: to set a random heading use the following line: Heading rnd 359)

5. Don’t forget to add a delay after this line to keep sphero moving in that direction for a half a second.

6. Have your program loop back to the initial color step after this delay. Be careful with where you loop back to because you don’t want to go all the way back to the first command where you set Sphero’s original heading of 0.

   Good Luck & Have Fun!

---

www.gosphero.com/education
Names:

Part 3 and 4:
In your own words explain what an accelerometer does:

What other common devices do you think have accelerometers in them?

What is the acceleration of an object in freefall? Include magnitude, direction, and the appropriate units.

Part 5:
Write down the code you built for the challenge: