

sphero

SPRK Lessons | STEM Challenges

Asociación Escuelas Lincoln | Buenos Aires, Argentina



Table of Contents

Introduction

SPRK Overview	3
Introduction to learning with Sphero.....	4

MacroLab Lesson 1 - Time, Speed, and Distance

Student Guide	6
Teacher Guide	12
Worksheet	16

MacroLab Lesson 2 - 2D Geometry

Student Guide	17
Teacher Guide	24
Worksheet	29

MacroLab Lesson 3 - Mean, Median, Mode

Student Guide	30
Teacher Guide	33
Worksheet	36

MacroLab Lesson 4 - Percentages

Student Guide	37
Teacher Guide	44
Worksheet	48

MacroLab Lesson 5 - Patterns and Color

Student Guide	49
Teacher Guide	55
Worksheet	59

OrbBasic Lesson 1 - Goto and Variables

Student Guide	60
Teacher Guide	67
Worksheet	70

OrbBasic Lesson 2 - If - Then - Else

Student Guide	71
Teacher Guide	77
Worksheet	80

OrbBasic Lesson 3 - Sensors and Random

Student Guide	81
Teacher Guide	86
Worksheet	89

STEM Challenges

Sphero Chariot	90
Sphero Bridge	94
Sphero Hydro-Hypothesis	98
Sphero Maze Mayhem	101

Play is a powerful teacher.

We create connected toys - but that's not all. By fusing technology with robotics, our toys are teaching and inspiring tomorrow's innovators and inventors. Programming isn't easy, but you don't need to be a rocket scientist to give kids a strong foundation. All you need is Sphero.

We created the SPRK program because we believe play is a powerful teacher. SPRK lessons give kids a fun crash course in coding while sharpening their skills in math and science.

CORE lessons

From polygons to percentages, CORE lessons explore the principals of math and science through programming. SPRK begins with MacroLab - a visual programming app. After completing 5 MacroLab lessons, students will be ready to take on lessons in OrbBasic - utilizing a powerful app for text-based programming. These 8 lessons align with the national Common Core Curriculum for 3rd, 4th and 5th grade.



STEM challenges

After completing CORE lessons, students can tackle STEM challenges. These 4 experiments foster creative problem-solving and teamwork. Students concept, engineer and build contraptions powered by Sphero and the programs they write as a group. After showcasing their creation to the class, an open-ended discussion is held, exploring which factors led some designs to out-perform others.



Hear what teachers
are saying on twitter:
[@SpheroEdu](https://twitter.com/SpheroEdu)

Introduction to learning with Sphero

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.

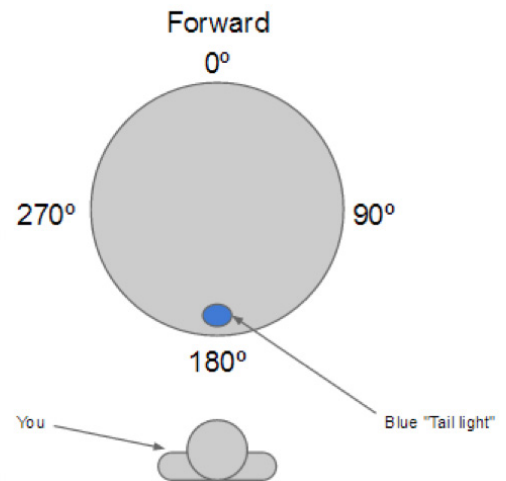
Help and support

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.

MacroLab Lesson 1

Time, Speed, and Distance: Student Guide

You are going to be using Sphero to figure out how time, speed, and distance relate to each other. Sphero can be programmed to roll at a certain speed for a certain amount of time. You are going to be creating programs to do just that, and then measure how far it goes. By changing the speed that Sphero rolls at and the time it spends rolling, you will be able to learn about how time, speed, and distance all relate.

The MacroLab commands you are going to use in this lesson are:



Roll – Makes Sphero roll at a given speed and heading.



Stop – Makes Sphero stop rolling immediately.

First you have to connect Sphero to the iPad (Part 1), then aim it (Part 2), and then there are two experiments (Part 3 and Part 4), and finally a challenge to see what you've learned (Part 4).

What can we figure out about time, speed, and distance?

If something goes at a certain speed for a certain amount of time, we know that it will go a certain distance. But how are these related? If an object moves faster, we know it will move farther, but do we know how much farther? Same for if it moves for a longer period of time.

The great thing about Sphero is that we can set its speed and we can set how long it should roll for. So then we just need to measure the distance. We can do that by marking the starting and ending points with masking tape, and measuring the distance between them.

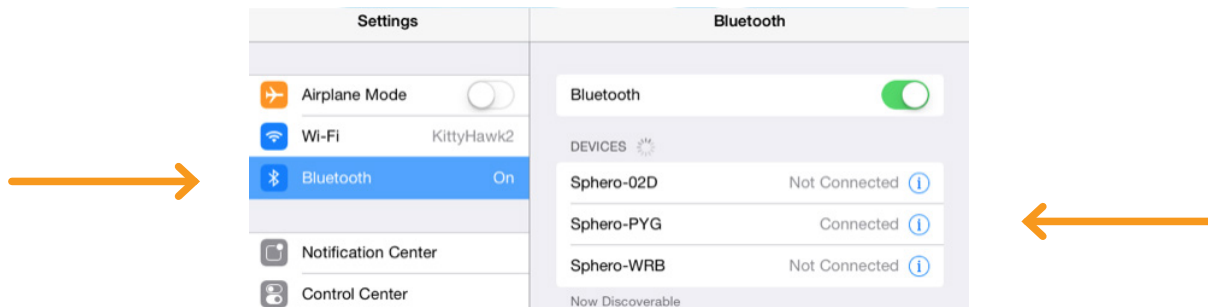
If we want to see how the distance has changed when we change the time or the speed, all we have to do is divide the new distance by the original distance. You'll do this math on your worksheet.

Part 1: Connect the Sphero

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

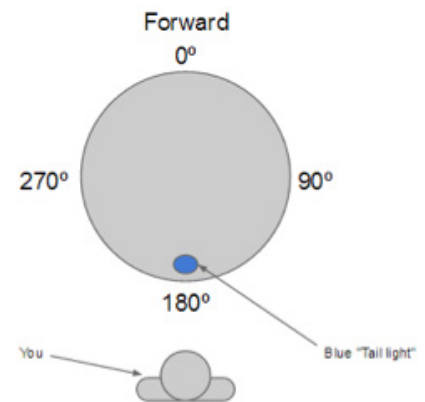
1. Pick up Sphero from its charging station and tap it twice on the logo to wake it up. You may have to tap it hard. It will start flashing colors when it is awakened out of its "sleep" state.
2. On your device, make sure Bluetooth is enabled. From the home page, click on Settings at the bottom. Then 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 1: Connect the Sphero (continued)



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 MacroLab.
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 MacroLab, 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!

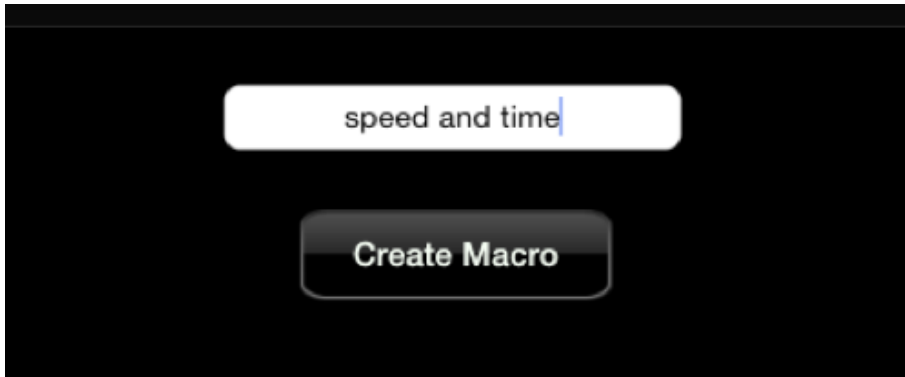
Part 3: Time and Distance

Now that we have Sphero going in the right direction, follow these steps for the first experiment:

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



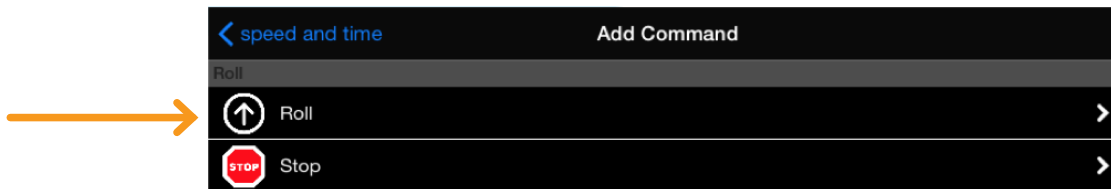
2. Where it says Macro Name, call type speed and time. Click Create Macro.



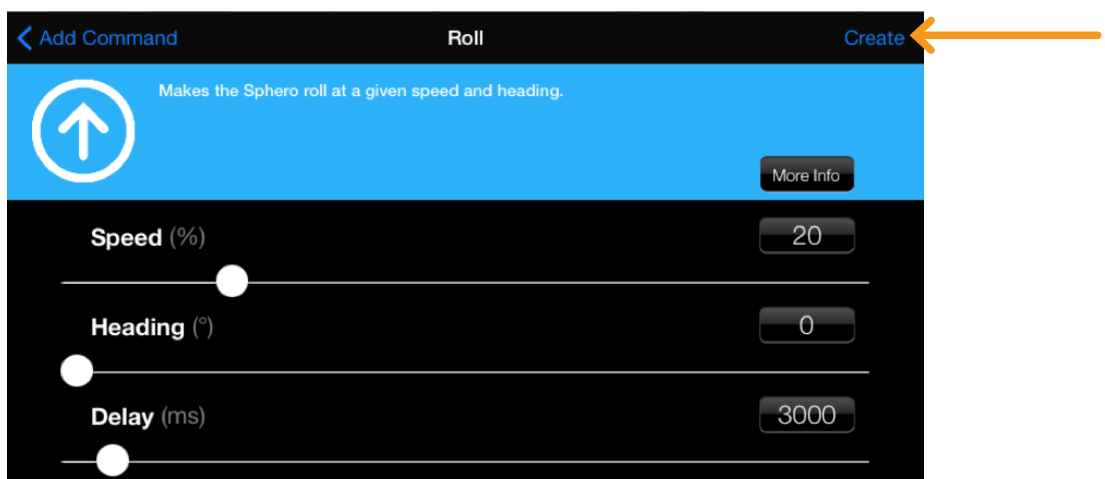
3. Add a command by tapping the + button at the bottom right.



4. Tap on Roll, the first command in the list.



5. Change the Speed to 20 and the Delay to 3000. It may be easier to use the keyboard than try to get the right values with the slider. Leave the Heading at zero. Click the Create button up top.

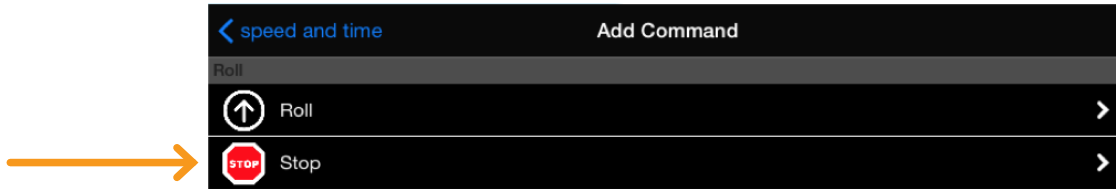


Part 3: Time and Distance (continued)

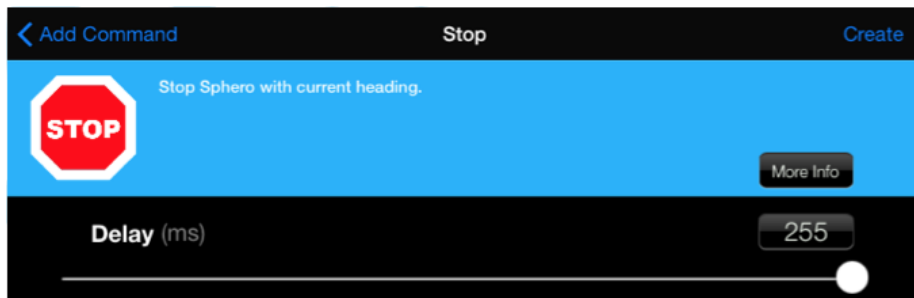
6. Add a new command using the plus key at the bottom.



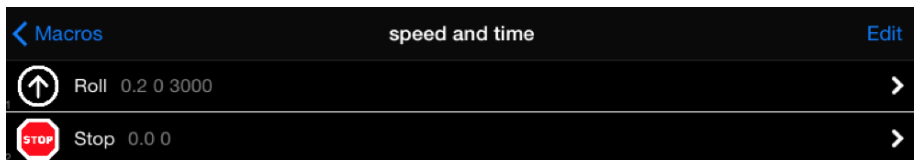
7. Choose Stop. This will stop Sphero from rolling immediately.



8. Move the bar all the way to the right to create a Delay of 255 and tap Create.



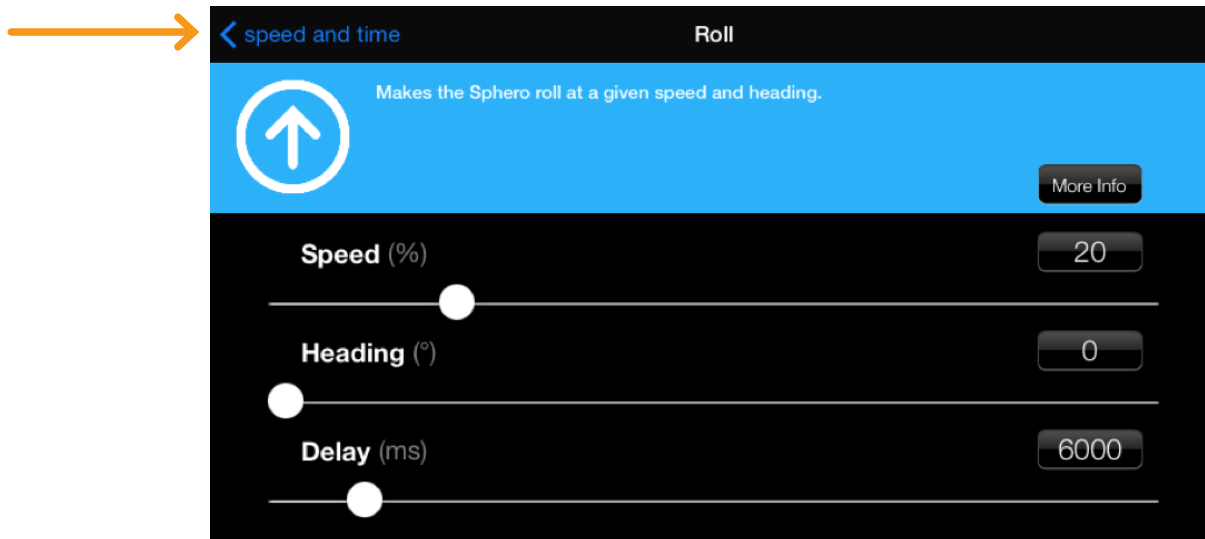
9. You've now written your first program! It tells Sphero to move at 20% speed for 3000 milliseconds, which means 3 seconds, and then stop. You will see a Roll command and a Stop Command. Click the Play button on the bottom.



10. The Sphero will roll slowly for 3 seconds.

Now that we have Sphero going in the right direction, follow these steps for the first experiment:

1. Put a small piece of masking tape on the floor. Place the Sphero on top.
2. On the device, tap Play. The ball will roll for 3 seconds. (If it doesn't roll the path that you want, you can aim Sphero again to be more accurate.)
3. With your tape measure, measure how far it traveled. Write the answer on your worksheet.
4. Now tap on the Roll line and change the delay to 6000. This will make it roll for 6 seconds. (To figure out how many seconds, divide by 1000.) Tap speed and time when you are done.



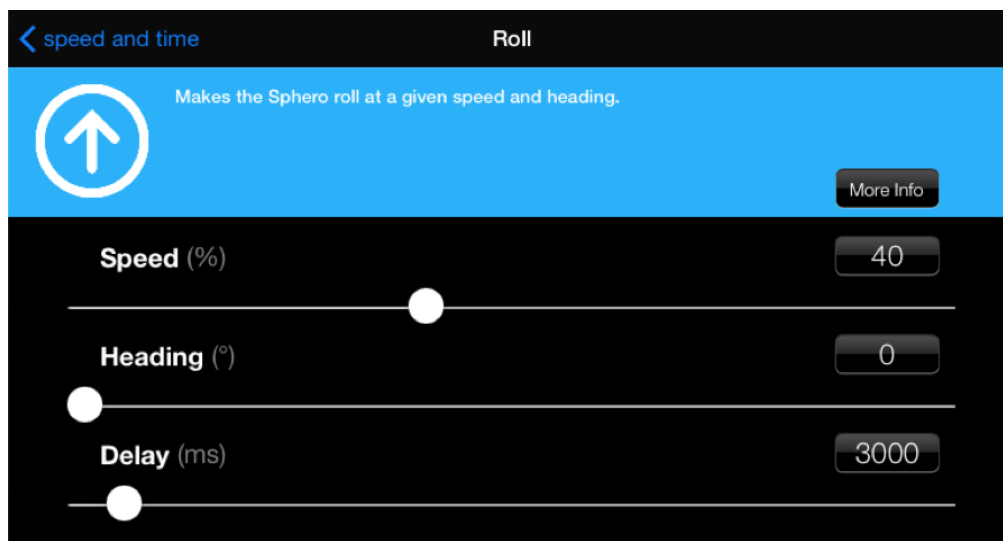
5. Put Sphero back on the tape and tap Play on the device. Measure the distance and write it on your worksheet.
6. Do this one more time with a delay of 9000 (9 seconds). Measure the distance and write it on your worksheet.
7. Either by hand or with a calculator (you can use the one on the iPad), divide the 6 second distance by the 3 second distance. Also divide 9 second distance by the 3 second distance. Write these numbers on the worksheet. (You only have to write one digit after the decimal point – that will be good enough.)

What do you notice about the distances and the time it took for Sphero to go those distances? Discuss this as a class.

Part 4: Speed and Distance

In our first experiment, we had Sphero move at the same speed, but we changed how long it was moving for. This time, we will have it move for the same amount of time, but we will change how fast it moves. Follow these steps:

1. Tap on Roll and change the delay time back to 3 seconds (a value of 3000.) Tap Play, and measure how far it goes. It should be about the same as the first time you measured it. Write it on your worksheet.
2. Now change the speed to 40%. Again, using the keyboard might be easier than the slider. Play, and measure how far it goes. Write it on your worksheet.



Part 4: Speed and Distance (continued)

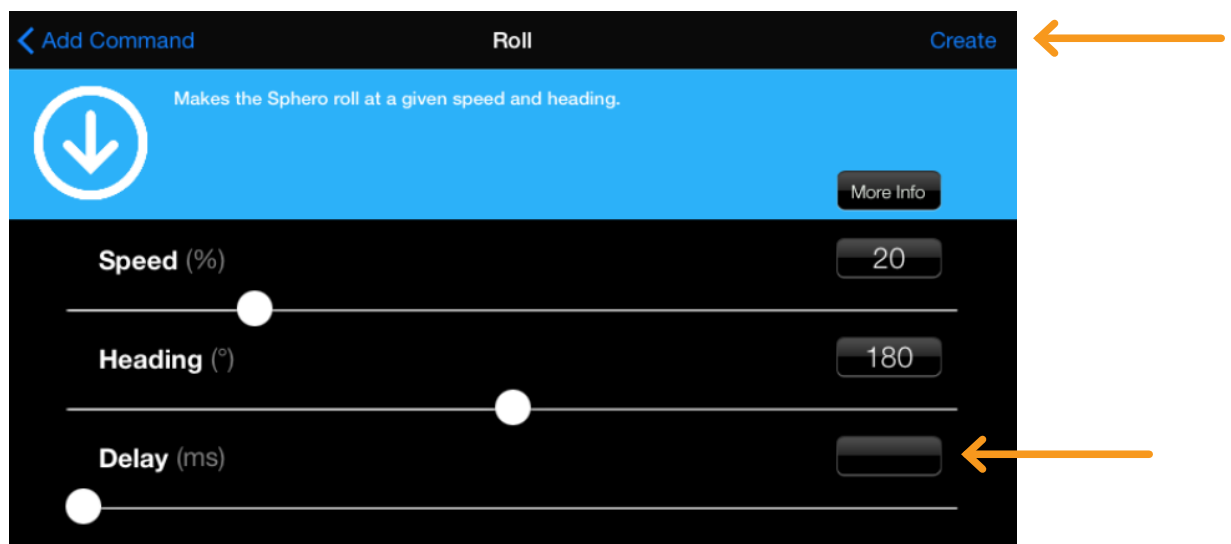
3. Now change the speed to 60%. Play, and measure how far it goes. Write it on your worksheet.
4. Either by hand or with a calculator, divide the 40% distance by the 20% distance. Also divide the 60% distance by the 20% distance. Write these numbers on your worksheet. (You only have to write one digit after the decimal point – that will be good enough.)

What do you notice about the distances and the speed the Sphero used to go those distances? Discuss this as a class.

Part 5: Challenge

For our challenge, we will have Sphero move a distance out, and then you have to figure out how to move it back at a given speed to have it stop about where it started. Follow these steps:

1. Tap the Roll line and change the speed to 40% and the delay to 5000. This will make it roll at 40% for 5 seconds. Tap the Back button to get back to the speed and time screen.
2. Tap the Add button at the bottom of the screen. Tap Roll. Set the speed to 20% and the heading to 180. This will turn Sphero around 180 degrees and start rolling back at 20% speed.
3. For the delay, put in a number that you think will bring it back to where it started. Tap the Create button up top.
4. Add another Stop command with a Delay of zero.



5. Tap the Play button. Does the Sphero come back about to where it started? If not, adjust the delay number on the second Roll by tapping it. It won't come back exactly, but it should come within a few inches. (If your surface is slippery, it might only be within about a foot.)

What number did you get? Why do you think that's the correct number?

MacroLab Lesson 1

Time, Speed, and Distance: Teacher Guide

Overview

Students will use Sphero to show that there is a linear relationship between time, speed, and distance. They will program Sphero to move at a particular speed for a particular amount of time, and then measure how far it has gone. They will use division to find the relationships between time, speed, and distance. Finally, they will be given a challenge to take what they've learned and to have Sphero return to the place where it started.

The mathematics could be made to be more complex (for example, students could derive a formula where they predict the distance). Read through the student guide. At the start of the lesson, go over the concepts of time, speed, and distance.

Objective

Students will:

- Create a one-line program that moves Sphero at a steady speed for a specified amount of time
- Perform measurements to determine the distance traveled.
- Perform division to compare different measurements
- Create a two-line program that moves Sphero to a certain position and then moves it back to where it started.

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.4.MD.A.2: Use the four operations to solve word problems involving distances, intervals of time, etc.
- 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 Macrolab loaded. You can get Sphero Macrolab for free from the iTunes app store.
- Sphero that has been fully charged
- Masking tape
- Tape measure
- Print-out of the worksheet (last page of teacher's guide)
- A flat clear path of at least 15 feet. (Preferably not very slippery.)

Part 1: Connect the Sphero

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

- Wake up the Sphero
- Turn on Bluetooth
- 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.

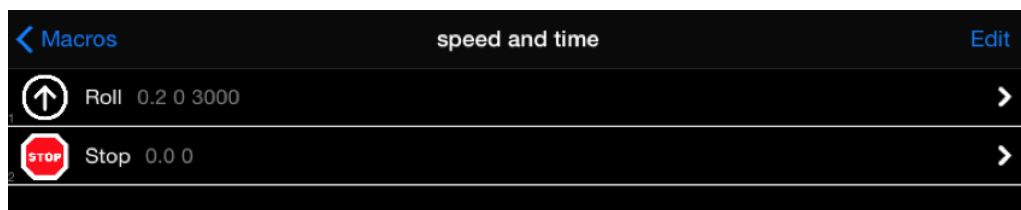
Part 3: Time and Distance

In part 3, students will create a macro (computer program for Sphero) and modify it in order to do an experiment. They will play the macro, measure how far the Sphero goes each time, and record the values in on their worksheet.

To create a macro to roll the Sphero, they will:

1. Create a new macro
2. Add a Roll command
3. Set the Roll command to move at 20% speed for 3000 milliseconds (3 seconds).
4. Add a Stop command to stop the Sphero immediately.

The final macro looks like this:



To run the experiment, they will:

1. Put a small piece of masking tape on the floor and put the Sphero on the tape.
2. Tap Play in MacroLab
3. Once the ball has moved, they will use the measuring tape to measure from the ball to the masking tape. They only need to be accurate within 1 inch or 2cm.
4. They will write the number down in the worksheet.
5. Then they will modify the macro so that it runs for 6 seconds and they will repeat the experiment, recording the result in the worksheet.
6. Finally, they will modify the macro so that it runs for 9 seconds and they will repeat the experiment, recording the result in the worksheet.
7. At the end they will divide the distance from the 6 second experiment by the 3 second experiment, and also the distance from the 9 second experiment by the 3 second experiment.

Ask them what they notice. They should be able to notice that when something travels at a particular speed, if it goes twice as long, it goes twice as far. If it goes three times as long, it goes three times as far. This is what's known as a linear relationship.

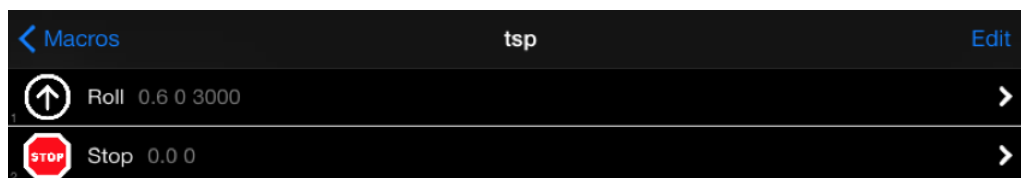
Why isn't it exactly twice or three times? The reason is that Sphero (or any real object) takes time to speed up and time to slow to a stop. This little bit of time at the beginning and end makes it so that it doesn't exactly travel the time we would expect it to.

Part 4: Speed and Distance

For part 4, the students will:

1. Modify the macro so that it once again rolls at 20% speed for 3000 milliseconds (3 seconds). They will run the experiment again and measure how far it goes, recording it in their worksheet. It should be close to what they had before.
2. Modify the macro so that it rolls at 40% speed for 3000 milliseconds. They will run the experiment again and measure how far it goes, recording it in their worksheet.
3. Modify the macro one last time so that it rolls at 60% speed for 3000 milliseconds. They will run the experiment again and measure how far it goes, recording it in their worksheet.
4. At the end they will divide the distance from the 40% experiment by the 20% experiment, and also the distance from the 60% experiment by the 20% experiment.

The final macro looks like this:



Ask them what they notice. They should be able to notice that when something travels for a particular speed, if it goes twice as fast, it goes twice as far. If it goes three times as fast, it goes three times as far.

Both the speed and time experiments show that distance is a linear factor of speed in time. In other words, we can find the distance traveled multiplying the speed times the time in motion. Right now, we are defining speed in terms of percent, but the actual speed that ball moves will be the distance traveled divided by the time it took to get there.

Part 5: Challenge

Part 5 is a challenge to see if the students have understood the concepts. For the challenge, students will have Sphero move a distance out, and then have to figure out how to move it back at a given speed in order to have it stop about where it started. They will:

1. Modify the Roll command to have it have a speed of 40% and a delay of 5000 milliseconds (5 seconds).
2. Add a new Roll command that has a speed of 20% and a heading of 180 degrees. The 180 degree heading is what will bring Sphero back in the direction it came.
3. Add a new Stop command to stop Sphero immediately.
4. Then they need to decide how long the delay time should be (that is, how long should it roll for). Let them try out different values until they get it right.

The final macro looks like this:



MacroLab Lesson 1

Time, Speed, and Distance: Worksheet

Names:

Part 3 - Time and Distance:

How far did the Sphero travel for Speed: 20% Time: 3000 (3 seconds): _____

How far did the Sphero travel for Speed: 20% Time: 6000 (6 seconds): _____

How far did the Sphero travel for Speed: 20% Time: 9000 (9 seconds): _____

What is the 6 second answer divided by the 3 second answer: _____

What is the 9 second answer divided by the 3 second answer: _____

Part 4 - Speed and Distance:

How far did the Sphero travel for Speed: 20% Time: 3000 (3 seconds): _____

How far did the Sphero travel for Speed: 40% Time: 3000 (6 seconds): _____

How far did the Sphero travel for Speed: 60% Time: 3000 (9 seconds): _____

What is the 40% answer divided by the 20% answer: _____

What is the 60% answer divided by the 20% answer: _____





Part 5 - Challenge:

What delay value did you use in the challenge: _____

MacroLab Lesson 2

2D Geometry: Student Guide

You are going to be using a Sphero to figure out what angle you need the Sphero to move at so that it can draw shapes with straight lines (also called polygons). The MacroLab commands you are going to use in this lesson are:

-  Roll – Makes the Sphero roll at a given speed and heading.
-  RGB – Changes the color of the Sphero.
-  Stop – Makes the Sphero stop rolling immediately.
-  Delay – Sets an amount of time the ball waits to execute the next command

First you have to set up the Sphero (Part 1), and then there are two shapes to draw (Square for Part 2 and Triangle for Part 3), and then a challenge to see what you've learned (Part 4).

How To Make Sphero Draw Shapes

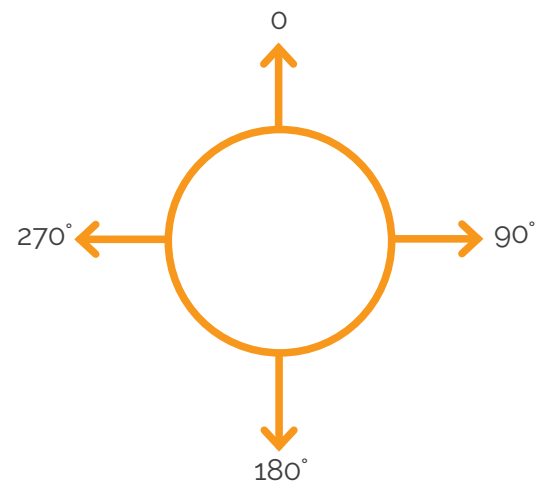
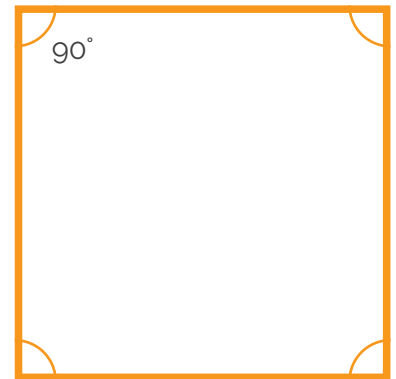
In this lesson, you're going to make Sphero trace the outline of shapes with straight sides, also known as polygons. Let's start with a simple shape, which is a square. A square has 4 equal sides, and the angle of each corner is 90 degrees.

So you might think that you should write a Sphero program like this:

1. Roll for 2 seconds
2. Turn 90 degrees right
3. Roll for 2 seconds
4. Turn 90 degrees right
5. Roll for 2 seconds
6. Turn 90 degrees right
7. Roll for 2 seconds

But this isn't how Sphero works. Sphero has a heading, which looks something like the figure to the right.

It will move directly away from you if the heading is zero degrees. To make it turn right, you need to set the heading to 90 degrees. To make it turn right again and head back towards you, you need to set the heading at 180 degrees. Don't think in terms of "turn right" or "turn left". Think instead about what degrees you want for the heading so that Sphero will move in the direction you want it to. For a square, those numbers are: 0, 90, 180, 270



How To Make Sphero Draw Shapes (continued)

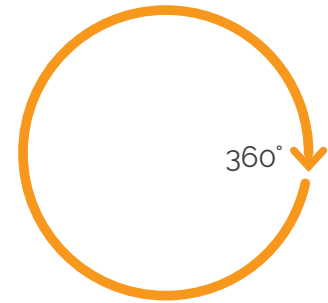
If you think about it, each Roll step, the heading number went up by 90.

$$90 = 0 + 90$$

$$180 = 90 + 90$$

$$270 = 180 + 90$$

Why 90? Well, the heading circle has 360 degrees. We want the Sphero to go full circle and end up where it started.



A square has 4 sides. Since each of those sides gets an equal share of the 360 degrees, each side gets an angle of:

$$360 / 4 = 90.$$

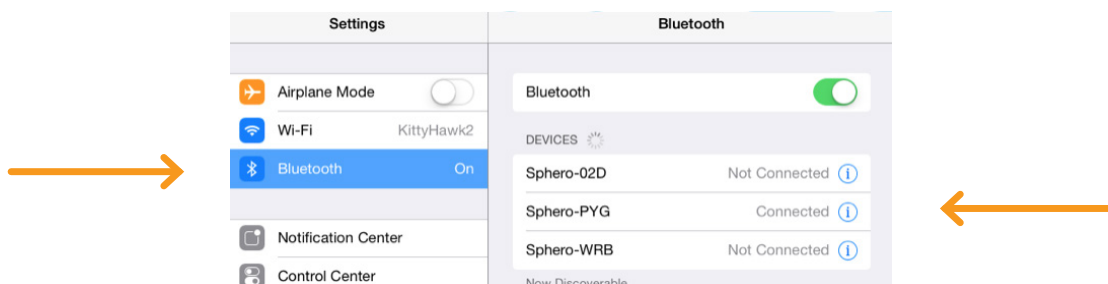
What if you wanted to draw a triangle instead? Then the heading angles would change by $360 / 3$, which is 120 degrees. The first angle would be 0 and the second angle would be 120 degrees. What would the third angle be?

You're going to be figuring this out, and more complicated shapes, too.

Part 1: Connect the Sphero

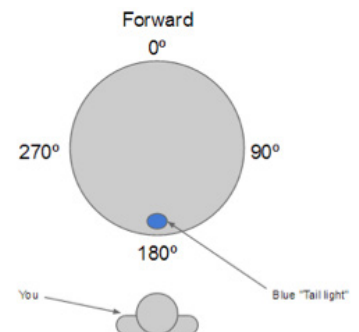
First thing you need to do is to connect the iPad to Sphero. Here's how:

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 MacroLab.
2. Have one of you hold the Sphero and stand at the beginning of the path you will use for your experiments. Have them face the direction the Sphero should roll, and hold the Sphero directly in front of them.
3. Now, you will aim the Sphero in that direction. Have a second member of the group use the iPad. In MacroLab, you will see a circle with two arrows at the bottom center of the screen. Tap on it and hold it.



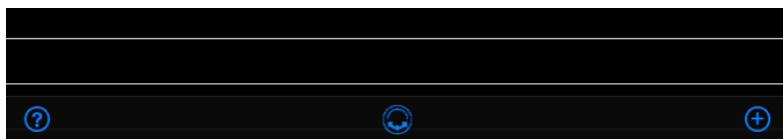
4. A white circle will appear. Move your finger slightly to rotate the insides of the Sphero. You will see a blue light inside the ball. Move it around until the blue light is directly facing the person holding the Sphero. This is the "taillight", and shows the direction opposite where the Sphero will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



Part 3: Square

Now that we have the Sphero going in the right direction, follow these steps to make the Sphero draw a square.

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



2. Where it says Macro Name, call it my square. Click Create Macro.

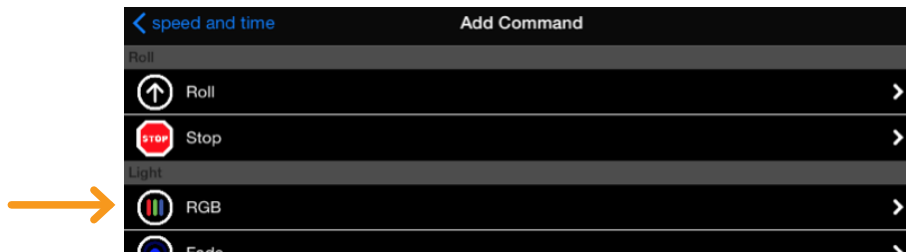


3. Add a command by tapping the + button at the bottom.

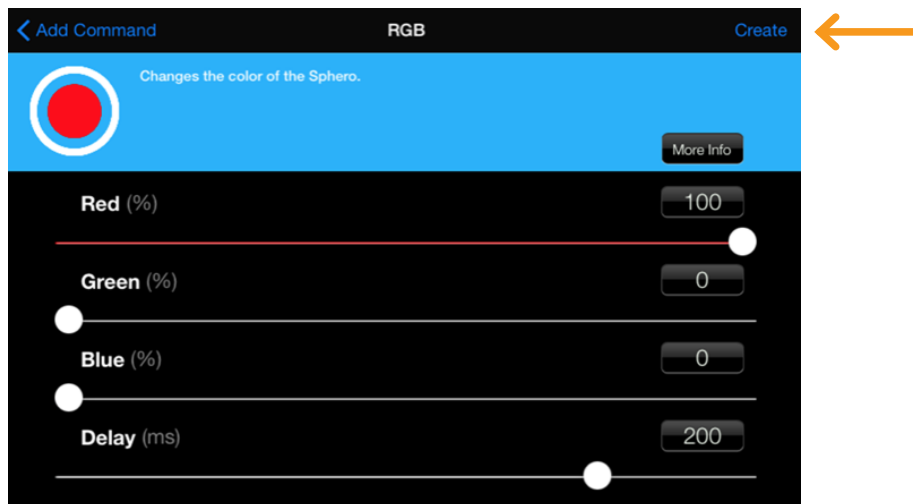


Part 3: Square (continued)

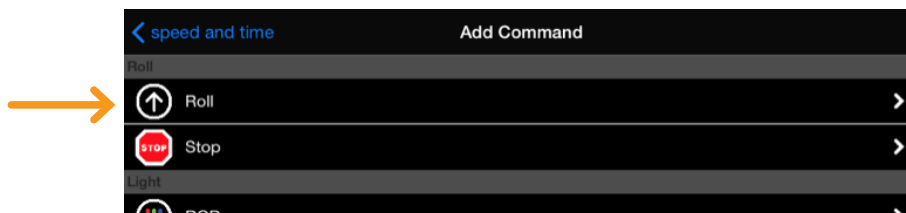
- For the fun of it, let's make every side of the square light up as a new color. So tap on RGB (which stands for "Red, Green, Blue"), the third command in the list.



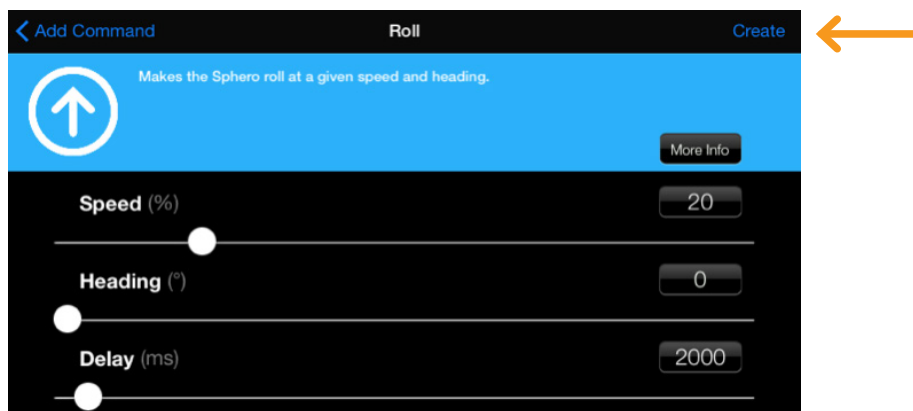
- Drag the slider for Red to the right for 100% in order to make a red color. Slide the Green and Blue sliders to the left for zero. Leave the Delay at 200. Click Create.



- Now let's add a Roll to create the first side of the square. Tap on Add to add a new command. Tap on Roll, the first command in the list.



- Set the Speed to 20 and the Delay to 2000 (2 seconds). It may be easier to use the keyboard than try to get the right values with the slider. Leave the Heading at zero. Click the Create button up top.



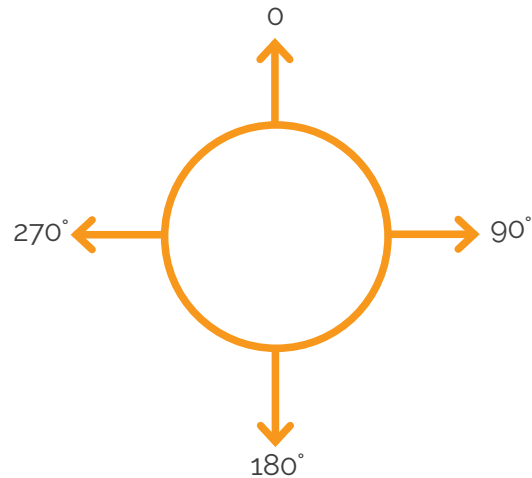
- Tap Play. You will see the Sphero turn red and go a short distance.

Part 3: Square (continued)

That's the first side of our square. Now, we want the Sphero to glow another color and make a right turn to make the second side. The color is easy:

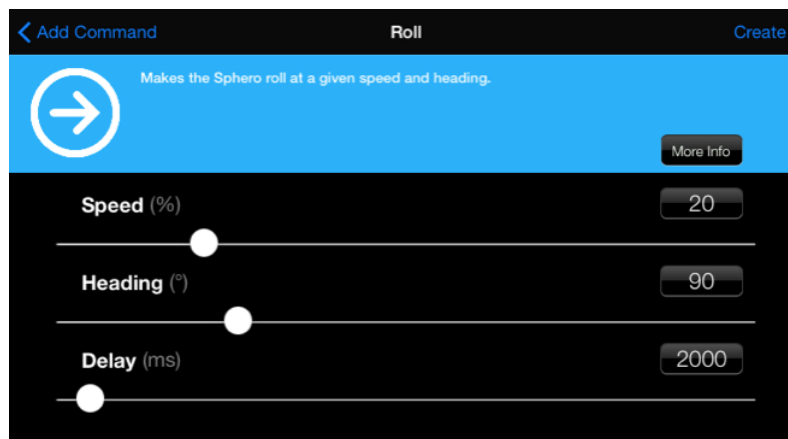
9. Add a third command by tapping the Add button. Tap RGB again, and this time bring the Green bar to 100% and the Red and Blue bar to 0%, making the color green. Click the Create button.

How do we get Sphero to turn? Remember, Sphero has a heading, which looks like this:



It will move directly away from you if the heading is zero degrees. To make it turn right, we need to set the heading to 90 degrees.

10. Add a fourth command by tapping the Add button. Chose Roll. Set the speed to 20%, the heading to 90, and the delay to 2000 (2 seconds). Tap Create.



11. Bring the Sphero back to the starting point and tap Play. The Sphero should create two sides of the square: red and green.
12. Add another command to turn the Sphero to blue. You should be able to figure out how to do this now.
13. Add a sixth command to have it create the third leg of the square. The speed and delay should be the same as the others, but the heading will be different. Take a look at the heading diagram to figure out what value it should be.
14. Move the Sphero back to the starting point and try it again. Do you have three sides of the square?
15. Lastly, let's add the color yellow. We can do this by mixing red and green together. (If you mix red and green paint, you'll get brown, but if you mix red and green light, you'll get yellow.) Create a new RGB command, and move the red and green bars to 100%, and the blue bar to 0%.

Part 3: Square (continued)



16. Now, add the last Roll command. Looking at the heading diagram, you should be able to figure out the heading.
17. Tap Play. Did you get a full square? If you did, congratulations!

Part 4: Triangle

In this section, you'll have the Sphero draw a triangle. How will you do this? If you remember back to the introduction, the heading angles for a triangle change by $360 / 3$, which is 120 degrees. Let's test this theory out and make a new macro that draws a triangle.

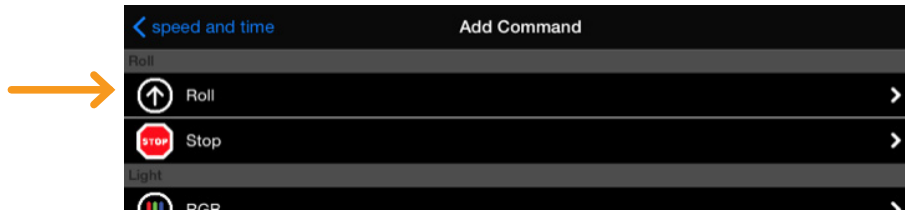
Follow these steps:

1. Return to the Macro screen by tapping on Macros in the top left corner.
2. Tap the + button at the bottom to create a new Macro
3. Name the Macro triangle.
4. Like before, add an RGB step for the color red.
5. Like before, add a Roll step that rolls at 20% speed, at 0 heading, and delay of 2000 milliseconds (2 seconds).
6. Again, add an RGB step for the color green.
7. Add another Roll step like the one you just did, but now we want the heading to be 120, since that's $360/3$.
8. Add an RGB step for the color blue.
9. Add one last Roll step with the same speed and delay. We want this step's heading to be 120 higher than the last one, or in other words, be $120 + 120$.
10. Put the Sphero at the starting point and tap Play to run the program. Did you get a triangle?

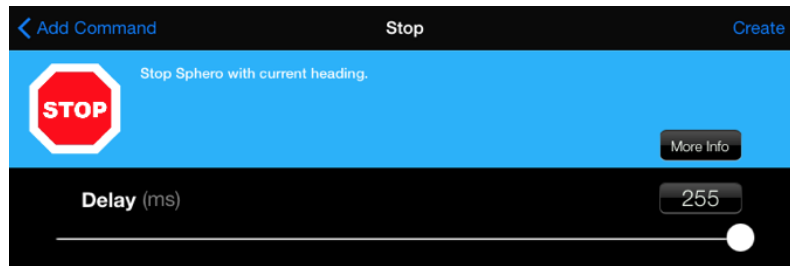
You may have noticed that the triangle's corners are a little rounded. That's because the Sphero has momentum when it comes to each corner. In other words, it had a little speed in the direction it was going in, and it doesn't change that speed right away. Let's fix that by adding two new commands between each of the Roll commands. These commands will stop the Sphero and make it wait before drawing the next leg of the triangle. Follow these steps:

1. Tap the + button at the bottom to add a new step.
2. Tap the Stop button.

Part 3: Triangle (continued)



3. Drag the delay to the maximum, which is 255. Click Create.



4. This will stop the Sphero, but we need it to sit for half a second more to give it time to find its new heading. Tap the + button again to add a new step.
5. Tap the Delay button. Change the value to 500 milliseconds (half a second). Tap Create.
6. Now we need to move these two steps into place under the first Roll command. Tap the Edit button in the top right corner.
7. Tap the three lines on the right side of the Stop step, and slide the step up until it is under the first Roll command.
8. Do the same for the Delay step, bringing it up under the Stop step. Then click Done in the top right. Your macro should look like this:



9. Tap Play. Does the first corner seem tighter than the second one?
10. Add another Stop step and Delay step with the same values as before. Tap the Edit button and slide them under the second roll. Tap Play and you should have tight corners for both your triangles.

Part 5: Challenge

For the challenge, see if you can create a new macro to draw a pentagon, which is a 5 sided shape. Call the macro "pentagon". Have the sides of the pentagon be red, green, blue, yellow, and purple. You don't need to put in Stop and Delay steps. Think about:

1. How many steps do you need?
2. What is the amount you will add to each heading to figure out the next heading?
3. What two colors will you mix to get purple?

MacroLab Lesson 2

2D Geometry: Teacher Guide

Overview

Students will use the Sphero robotic ball to draw polygons and learn the relationship between heading angles and the number of sides in a polygon where all sides are equal. They will also learn how to set the colors of the lights inside the Sphero.

Read through the student guide. At the start of the lesson, go over the concepts of heading and what angles are used to draw a square and a triangle.

Objective

Students will:

- Learn about how the Sphero has a heading that goes from 0 degrees to 360 degrees, and that it determines in which direction it will roll.
- Students will write a program to make Sphero draw a square, where each side of the square is a different color.
- Students will write a program to make Sphero draw a triangle, determining the heading angles from clues.
- Students will write a program to make Sphero draw a pentagon, determining the heading angles from a formula.

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.4.NBT.B.6: Find whole-number quotients and remainders with up to four-digit dividends and one-digit divisors, using strategies based on place value, the properties of operations, and/or the relationship between multiplication and division.
- CCSS.MATH.CONTENT.4.G.A.2: Classify two-dimensional figures based on the presence or absence of parallel or perpendicular lines, or the presence or absence of angles of a specified size.
- CCSS.MATH.CONTENT.4.MD.C.5.A: An angle is measured with reference to a circle with its center at the common endpoint of the rays, by considering the fraction of the circular arc between the points where the two rays intersect the circle.
- CCSS.MATH.CONTENT.4.MD.C.7: Recognize angle measure as additive.
- CCSS.MATH.CONTENT.5.OA.B.3: Analyze patterns and relationships
- CCSS.MATH.CONTENT.5.G.B.3: Understand that attributes belonging to a category of two-dimensional figures also belong to all subcategories of that category.
- CCSS.MATH.CONTENT.5.G.B.4: Classify two-dimensional figures in a hierarchy based on properties.
- 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 iPad, but other devices could be used. Here is what each group would need:

- iPad with Sphero MacroLab loaded. You can get Sphero MacroLab for free from the iTunes app store.
- Sphero that has been fully charged
- A flat clear area of at least 8 feet by 8 feet. (Preferably not very slippery.)

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: Square

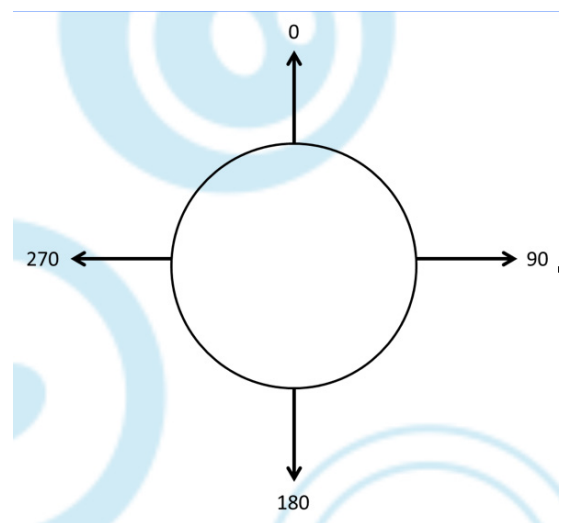
In part 3, students will create a macro (computer program for the Sphero) and add steps to it to draw a square with each side a different color.

You may need to discuss the heading diagram for the Sphero.

This shows what heading values you should use to direct the Sphero to roll in a particular direction. Note that this diagram only shows 90-degree angle increments, but that you can have angles anywhere in between.

To create a macro to draw the square, they will:

1. Create a new macro.
2. Add a color (RGB) command and a Roll command.
3. Add another RGB command and a Roll command, this time with a 90-degree heading.
4. Figure out that they need to add another RGB command and a Roll command, this time with a 180-degree heading.
5. Figure out that they need to add another RGB command and a Roll command, this time with a 270-degree heading.



Part 3: Square (continued)

Note: they will try out the macro at various points as they are creating it so that they can see what each step does. The final macro will look like this:



Part 4: Triangle

For part 4, students will modify the macro to draw a triangle. You should discuss why there are 90-degree increments for the square: the 360 degrees in a circle is divided into 4 equal parts. Now, for a triangle, they will divide it into 3 equal parts, each with 120 degrees. The first heading angle is zero, the second one is 120 degrees, and the third one is 240 degrees ($120 + 120$).

The students will:

1. Create a new macro.
2. Add a color (RGB) command, a Roll command with a 0 heading.
3. Add another RGB command and a Roll command, this time with a 120-degree heading, and a Stop command with delay of 255.
4. Add another RGB command and a Roll command, this time with a 240 degree heading, and a Stop command with delay of 255.

The full macro is:



Part 4: Triangle (continued)

Then, to make the corners crisper, the students will put Stop and Delay commands at the end of each side of the triangle.

The full macro is:













Part 5: Challenge

Part 5 is a challenge to see if the students have understood the concepts. For the challenge, students will have the Sphero draw a pentagon. To do this, they need to figure out that a pentagon has 5 sides, of equal length and therefore the heading directions will be spaced out by $360 / 5 = 72$ degrees. Then they need to create the macro as follows:

1. Add an RGB command for red and a Roll with a heading of 0.
2. Add an RGB command for green and a Roll with a heading of 72.
3. Add an RGB command for blue and a Roll with a heading of 144. ($72 + 72$)
4. Add an RGB command for yellow and a Roll with a heading of 216. ($144 + 72$)
5. Add an RGB command for purple and a Roll with a heading of 288. ($216 + 72$)

Optionally, they can put Stop commands with delay after each Roll, which will make the corners crisper. The full macro can be found on the following page:

Part 5: Challenge (continued)

Macros		challenge	Edit
1	 RGB 1.0 0.0 0.0 200		>
2	 Roll 0.2 0 1000		>
3	 RGB 0.0 1.0 0.0 200		>
4	 Roll 0.2 72 1000		>
5	 RGB 0.0 0.0 1.0 200		>
6	 Roll 0.2 144 1000		>
7	 RGB 1.0 1.0 0.0 200		>
8	 Roll 0.2 216 1000		>
9	 RGB 1.0 0.0 1.0 200		>
10	 Roll 0.2 288 1000		>

MacroLab Lesson 2

2D Geometry: Worksheet

Names:

Part 3 - Square:

What are the four heading angles you used in your macro: 0, 90, _____, _____

What are the four colors you used in your macro?

RED: 100% red, 0% green, 0% blue

GREEN: _____% red, _____% green, _____% blue

BLUE: _____% red, _____% green, _____% blue

YELLOW: _____% red, _____% green, _____% blue

Part 4 - Triangle:

What are the three heading angles you used in your macro? _____ _____ _____

Part 5 - Challenge:

What are the five heading angles you used in your pentagon macro? _____ _____ _____ _____ _____

What are the five colors you used in your Macro?

RED: _____% red, _____% green, _____% blue

GREEN: _____% red, _____% green, _____% blue

BLUE: _____% red, _____% green, _____% blue

YELLOW: _____% red, _____% green, _____% blue

PURPLE: _____% red, _____% green, _____% blue

MacroLab Lesson 3

Mean, Median, & Mode: Student Guide

This lesson builds on the previous 2 in a more open-ended, experiment-based way. You are going to be using Sphero to knock over an object that your teacher has set up a few feet away. In this lesson you will use division to see how much delay is needed for sphero to reach a target at different speeds. At the end of the lesson you will use Sphero to knock over whiteboard markers, or other objects in a bowling challenge that explores Mean, Median, and Mode. The Macrolab commands you are going to use in this lesson are:

 Roll – Makes the Sphero roll at a given speed and heading.

First you have to connect Sphero to the iPad (Part 1), then aim it (Part 2), and then there are two activities (Part 3 and Part 4), and finally a challenge to see if you can apply what you've learned (Part 5).

Review of Milliseconds, Rate, Distance, and Time:

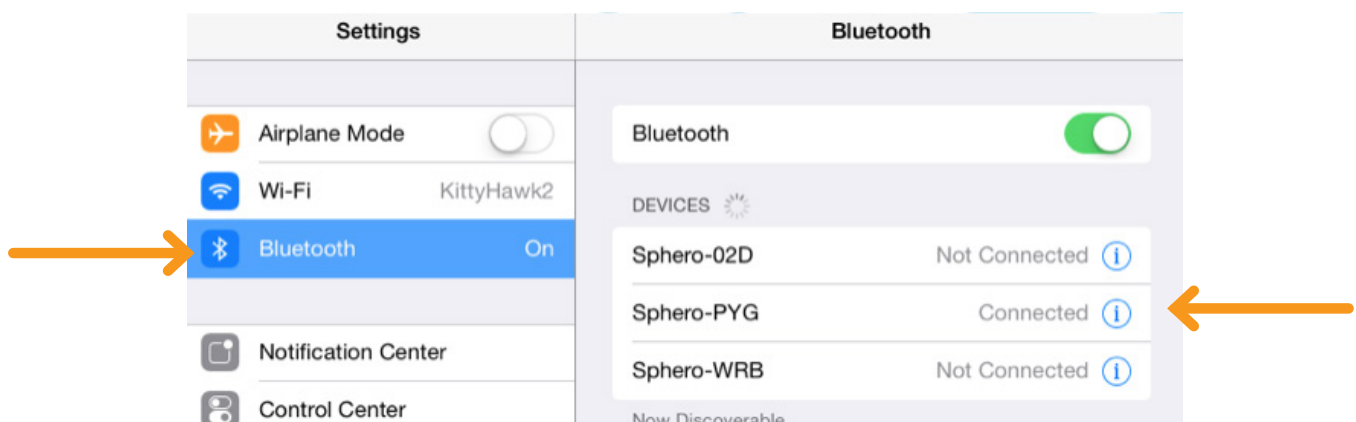
Remember from MacroLab Lesson 1 that Sphero delays are programmed in milliseconds, or 1/1000 of a second, so to roll for 1 second you will use delay of 1000 ms. Also from MacroLab Lesson 1, rate, time and distance are all related and you can calculate any one of them using the other two.

Rate (speed) = Distance / Time (delay)

Part 1: Connect the Sphero

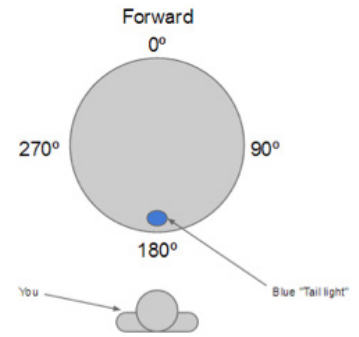
First thing you need to do is to connect the iPad to Sphero. Here's how:

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.



Follow these steps to aim the Sphero:

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

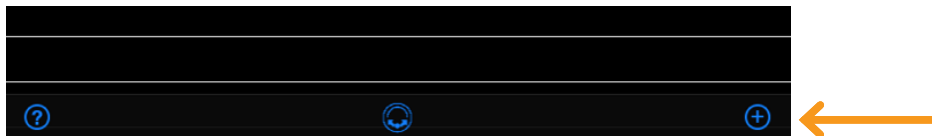


4. A white circle will appear. Move your finger slightly to rotate the insides of the Sphero. You will see a blue light inside the ball. Move it around until the blue light is directly facing the person holding the Sphero. This is the "taillight", and shows the direction opposite where the Sphero will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



Part 3: Reaching the Target

1. Now that we have the Sphero going in the right direction, follow these steps to reach the object that your teacher set up for you to knock over such a whiteboard marker. Tap the + button at the bottom to create a new macro.



2. Where it says Macro Name, call it Bowling. Click Create Macro.

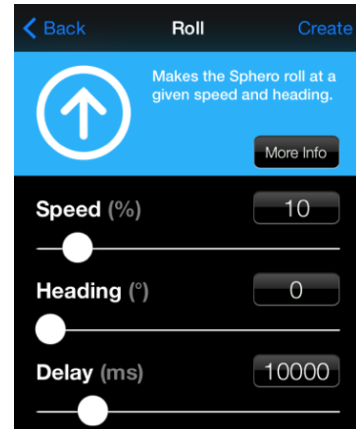


Part 3: Reaching the Target (Continued)

3. Add a new command by clicking the + button.
4. Choose Roll

Using only 10% speed, test different delays to see what the delay should be so that Sphero knocks over the target object and stops within 6 inches of that object.

5. To do this set the speed to 10% and the delay to a value of your choice, keep the heading at 0 degrees. Click Create.
6. Record on your worksheet each delay value that you try.
7. Record on your worksheet the delay value that was successful in knocking over the object.



Part 4: Calculations

Once you have identified the right amount of delay it is time to make some calculations to figure out how long of a delay will be needed to reach the same target when moving at higher speeds.

1. Now that you know the delay needed at 10% speed, use math to figure out how much delay is needed to reach the object when moving at higher speeds.
2. Fill in the table in your worksheet, there is an example calculation already done for you for 20% speed
3. Test out your math and run trials to see if you can knock over the object at 30%, 50%, 80%, and 100% speeds. Only do 3 trials for each speed and see if you can be successful at all speeds.

Part 5: Challenge

As a group you will have 5 tries to knock over all 5 whiteboard markers (or other object used for pins).

1. As a team, determine what speed and delay you want to use to knock over as many objects as possible
2. Once you are ready, let the teacher know you want to run your tests. You will have 5 minutes to run your trials; you can make adjustments after each run.
3. Keep track of how many markers you knock down for each trial on your worksheet.
4. Once you have these trials complete and have recorded your results add them to the master list for the class
5. Once the master list is complete, on your worksheet record all the trial results in order from lowest to highest.
6. Determine the Mean, Median, and Mode of this data set.

MacroLab Lesson 3

Mean, Median, & Mode: Teacher Guide

Overview

Students will use Sphero to conduct an experiment. They will identify how long they need Sphero to move at 10% speed to reach and knock over a target object (such as a whiteboard marker). Using this information, they will then calculate how long Sphero will need to move at higher speeds to reach the same target. Finally, they will be given a bowling challenge where they can use any speed they like to knock over as many "pins" (whiteboard markers) as they can. In the challenge, they will record their results and then they will determine the mean, median, and mode of the results from the entire class.

Read through the student guide. At the start of the lesson, review the relationship between time, speed, and distance. Also introduce mean, median, and mode.

Objective

Students will:

- Students will solve an open ended problem with guess and check.
- Students will use division and the relationship between rate, distance, and time to determine the time needed to reach a target
- Students will calculate mean, median, and mode from a data set that they helped generate

Common Core Math Standards

The following Common Core Math Standards for 4th, 5th and 6th grade apply to this lesson:

- CCSS.MATH.CONTENT.6.SP.A: Develop understanding of statistical variability
- CCSS.MATH.CONTENT.6.SP.B: Summarize and describe distributions
- CCSS.MATH.CONTENT.4.OA.A: Use the four operations with whole numbers to solve problems.
- 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 by mobile devices, either Apple (iPhone or iPad) or Android. 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 iPad, but other devices could be used. Here is what each group would need:

- iPad with Sphero MacroLab loaded. You can get Sphero MacroLab for free from the iTunes app store.
- Sphero that has been fully charged
- A flat clear area of at least 8 feet by 8 feet. (Preferably not very slippery.)
- Masking tape
- 5-7 Whiteboard Markers or other object students can knock over with Sphero

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 MacroLab 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: Reaching the target

Set up the target object 7 feet away from a starting line. You can use masking tape on the floor to mark the starting line.

To begin this portion, let the students have a few practice rounds to ensure Sphero is heading straight towards the target.

Now that the students have Sphero moving straight, they need to figure out when Sphero is set to 10% speed how long of a delay they need to knock over the target object and stop within 6 inches of it.

Part 4: Calculations

Once they have determined the correct delay to knock over the target, then they need to figure out how long of a delay to use for Sphero being driven at 30%, 50%, 80%, and 100% speeds. Students may need to use the calculator on the I-pad, otherwise they should be able to complete all of the calculations using long division. Students then should fill in the table on their worksheet.

% speed	10	20	30	40	50	60	70	80	90	100
delay (ms)	10,000	5,000	3,333	2,500	2,000	1,667	1,429	1,250	1,111	1,000

Example of solution (note: for your reference this table includes all delays, not just the 30,50,80, and 100% speeds):

Part 4: Calculations (continued)

Here is a simple approach for students to solve for the delays of different speeds:

If 10% of speed needs 10,000 ms to reach the target:

Distance= Speed*Delay

Distance= 10*10,000

Distance= 100,000

For 20%, we still need to go the same distance (this one is straight forward because the speed has doubled but just to outline the process)

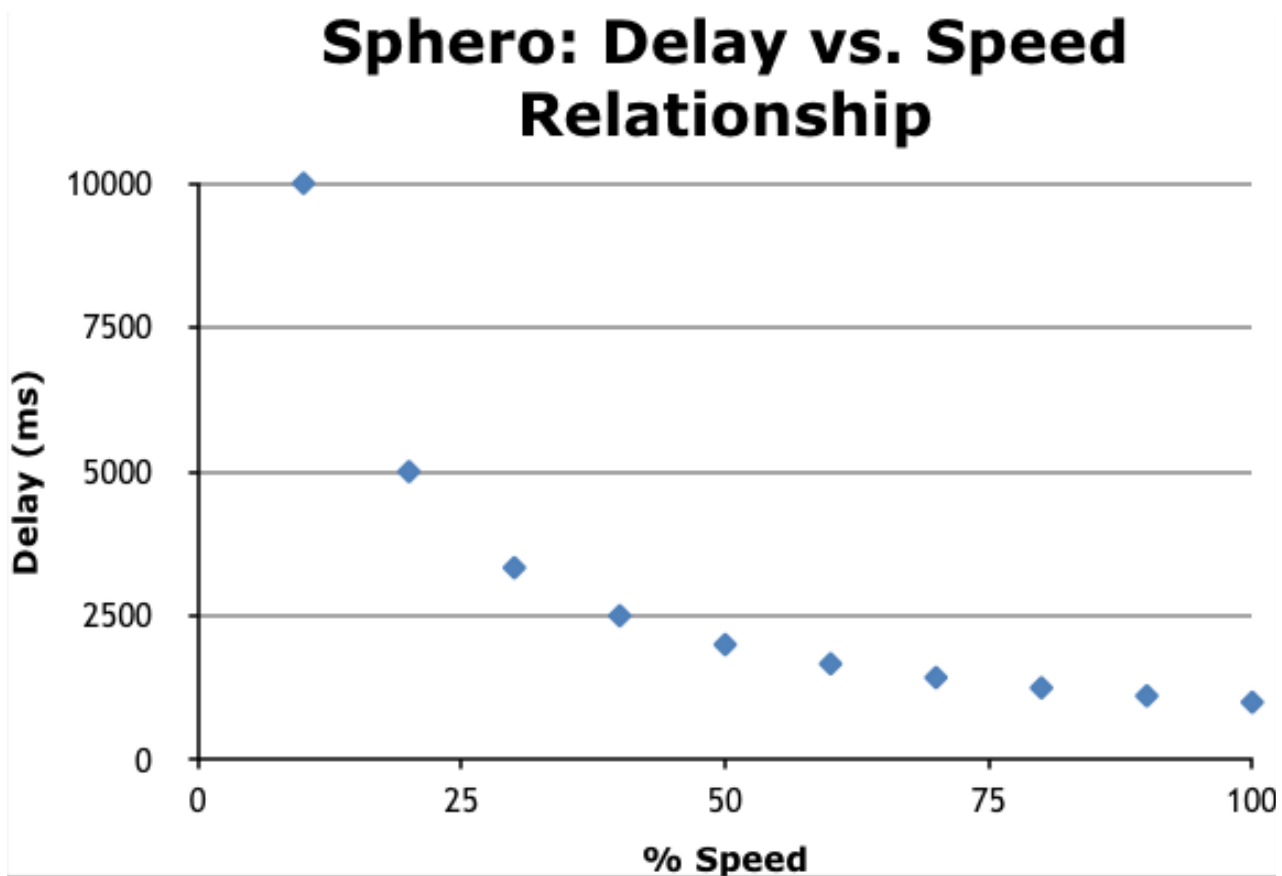
Distance= Speed*Delay

100,000= 20*delay

100,000/20= delay

5,000

While not included in the student guide, you may find it may be interesting to have students plot the Delay vs. Speed relationship and speak about the shape of the graph.



At the end of this section, there is time for students to test their math and see how close they can get to the target at each speed. They are prompted to take a few trials at 30%, 50%, 80%, and 100% to see if they can successfully knock over the target using the delay value they calculated.

Part 5: Challenge

For this challenge, set up 5 whiteboard markers at the same target location used earlier in the lesson. This challenge is a fun game of Sphero bowling where students determine which speed and delay they would like to use to knock over as many of the whiteboard markers over as they can. They need to record how many markers they knock over each of their 5 tries. Then, as a class compile the results of all trials and students will determine the mean, median, and mode of this data set.

MacroLab Lesson 3

Mean, Median, & Mode: Teacher Guide

Names:

Part 3:

List the delays are you trying to just reach the target at 10% speed:

What delay worked best to reach the target?

Delay	Result
(Example 20,000 ms)	(Moved too far past target)

Part 4:

Using the delay value you found and the fact that Sphero is moving at 10% speed can you figure out what the delay should be at 30%, 50%, 80% and 100%. to reach the same target.

Remember: Delay X speed = distance

% speed	10	30	50	80	100
delay (ms)					
delay that knocked over object (ms)					

Part 5:

What speed and delay will you use to knock over the pins? _____

Trial #	1	2	3	4	5
Pins knocked over					

List all bowling results from the class, from lowest to highest:

What is the median?




What is the mode?

What is the mean?

MacroLab Lesson 4

Percentages: Student Guide

You are going to be using Sphero to learn about percentages. The Sphero uses percentages in two places: speed and color. You will be using these Sphero macro commands:

-  Roll – Makes the Sphero roll at a given speed and heading.
-  RGB – Changes the color of the Sphero.
-  Delay – Sets an amount of time the ball waits to execute the next command

First you have to connect the Sphero to the iPad (Part 1), then aim your Sphero in the correct direction (Part 2). Next you will perform an experiment and challenge for speed (parts 3 and 4), and an experiment and challenge for color (parts 5 and 6).

What is Percentage?

The word “percent” comes from the Latin “per centum”, meaning “by 100”. When we say 50% (50 percent), which means 50 out of 100, which can be expressed as a decimal or a fraction:

$$50\% = \frac{50}{100} = 0.50 = \frac{1}{2}$$

When you see a percentage, to figure out what decimal or fraction it is, you just need to divide it by 100. And remember, to divide by 100, all you need to do is to move the decimal point by two digits.

How Does Sphero Use Percentages?

For Sphero, the speed is expressed as the percentage of the fastest speed it can roll. So 100% speed means that fastest it can go, and 50% speed means half that speed.

For colors, Sphero has three colored lights: red, green, and blue. 100% red means it lights up the red light as much as it can and 50% means that it lights it up half as bright. The same is true for green and blue. But what’s cool is that you can mix different colors to create new colors, and percentages become important then.

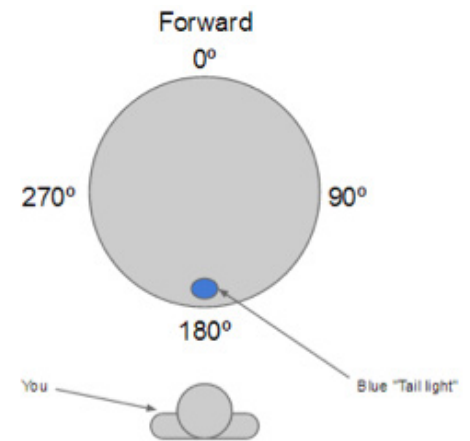
Part 1: Connect the Sphero

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.

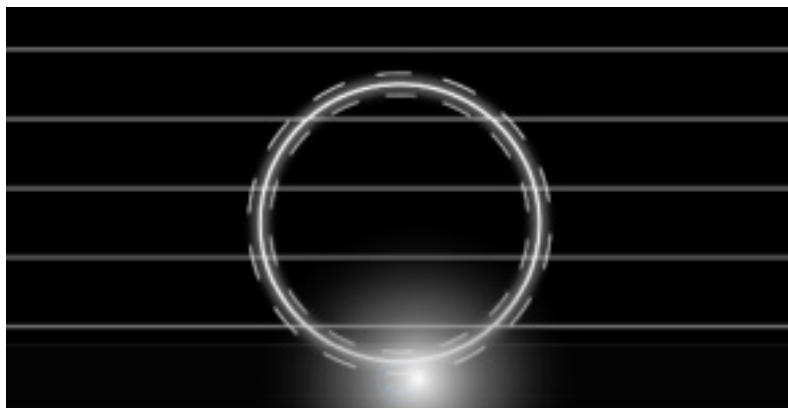


Follow these steps to aim the Sphero:

1. Go to the home screen and open MacroLab.
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 MacroLab, 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.

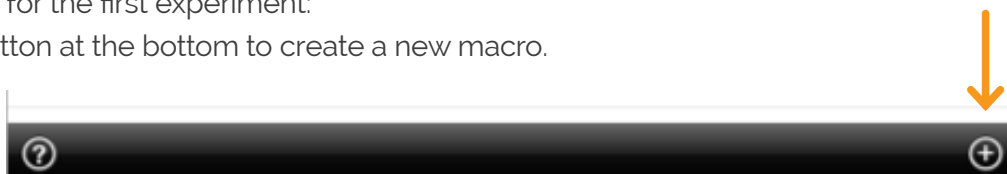


Important: For these experiments, the Sphero will travel a long distance, so be sure to aim the Sphero as accurately as you can to keep it on track. You can also re-aim Sphero anytime!

Part 3: Speed

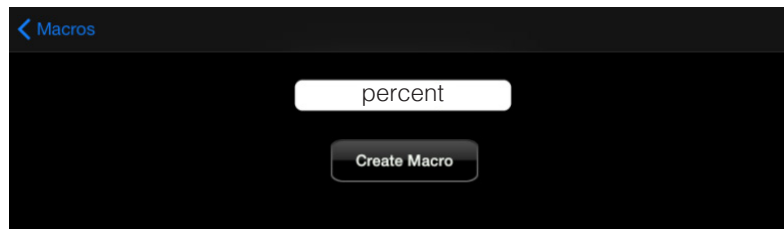
Now that we have Sphero going in the right direction, let's see what the different speed percentages do. Follow these steps for the first experiment:

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



Part 3: Speed (continued)

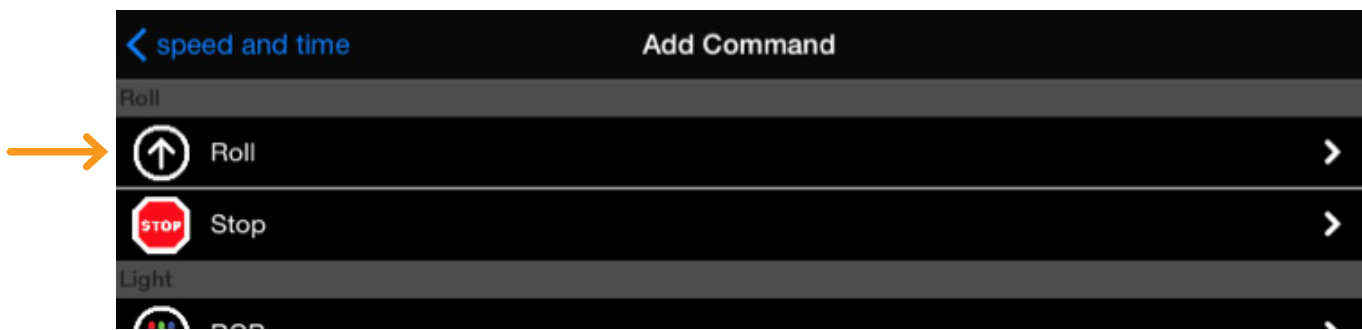
- Where it says Macro Name, call type percent. Click Create Macro.



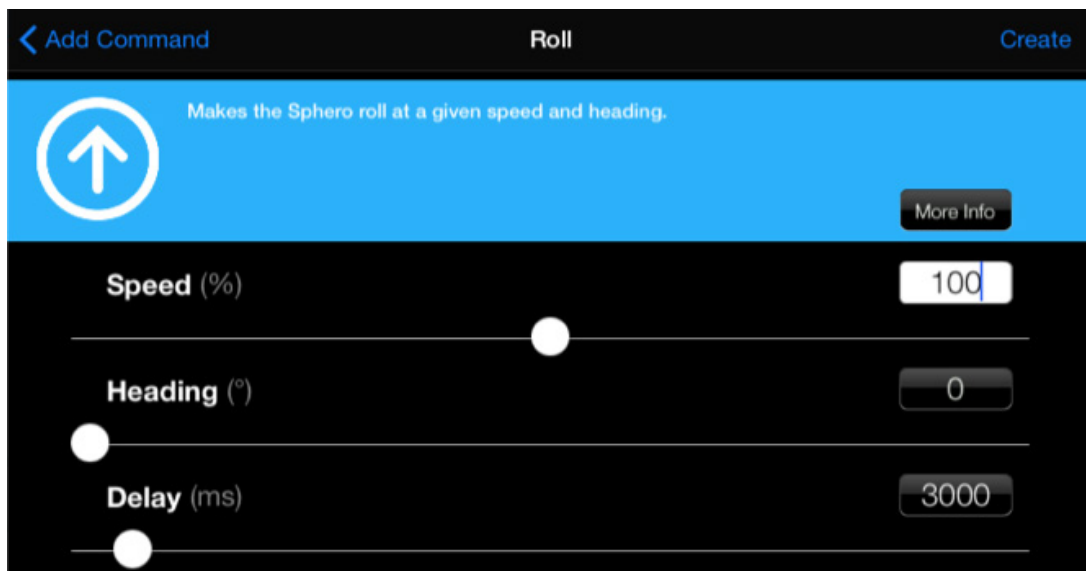
- Add a command by tapping the + button at the bottom right.



- Tap on Roll, the first command in the list.



- Change the Speed to 100 and the Delay to 3000. Leave the Heading at zero. Click the Create button up top.

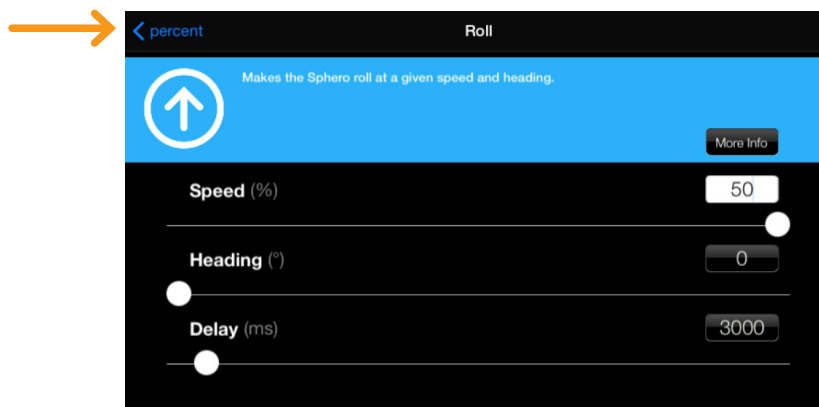


- Your program now moves Sphero at 100% speed for 3000 milliseconds, which is 3 seconds. You will see a Roll command.
- Tap the Play button. Make sure your Sphero can roll freely for the entire three seconds. You may have to adjust the direction like you did in Part 1 (Setup) to get it on right path.

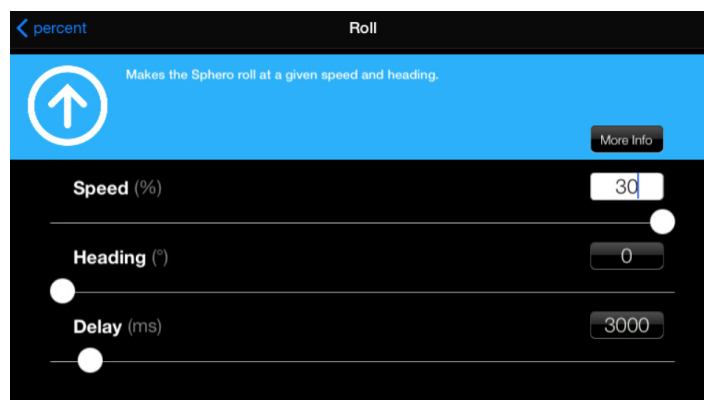


Now, let's do the experiment! Follow these steps:

1. Put a small piece of masking tape on the floor. Place Sphero on top.
2. Have one member of the team hold the iPad, and the rest of the team evenly spaced along the path the Sphero will roll. Have the person with the iPad tap the Play button to start Sphero. Everyone else should be looking for when Sphero stops rolling. At this high speed, once it stops rolling, it will continue to slide, but you want to make sure you know where it stopped rolling.
3. Place a small piece of masking tape where Sphero stopped rolling.
4. With your tape measure, measure how far it traveled. Write the answer on your worksheet.
5. Let's calculate the actual speed of Sphero. The speed is the distance traveled divided by the time it took the travel it. So, divide the distance by 3 seconds, and write that in your worksheet.
6. Now, let's do it again for 50% speed. Tap on the Roll line and change the speed to 50%. Tap percent when you are done.



7. Put Sphero back on the starting piece of tape and tap Play on the device. Once again, put a piece of tape where Sphero stops rolling. Measure the distance between the starting and ending pieces of tape and write it on your worksheet.
8. Do this one more time with a speed of 30%. Measure the distance and write it on your worksheet. Divide the distance by 3 seconds to get the speed, and write that on your worksheet, too.



9. Now, for our final calculations. Divide the 50% speed by the 100% speed. Turn this number into a percentage. You should have a number somewhere near 50. Divide the 30% speed by the 100% speed, and turn this into a percentage. You should get a number close to 30.

Why is it not exactly 50 and 30? It turns out that the speed calculation is not exact because Sphero takes a little time to get up to speed. The faster the speed, the more time it takes to go from zero to its final speed.

Part 4: Speed Challenge

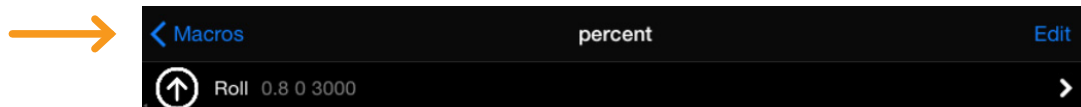
For the challenge, let's work backwards and see if we can figure out what speed percentage we need to get Sphero to stop rolling where we want to. Follow these steps:

1. Put a piece of tape somewhere between your starting piece of tape and your ending piece of tape for the 100% speed. Make it closer to the end than the beginning.
2. Measure the distance from the starting tape to the new tape you put down. Write that number in your worksheet.
3. If Sphero only has 3 seconds, how fast does it need to go to stop rolling at that point? Speed is distance divided by time, so divide the distance you measured by 3 seconds. Write the answer in your worksheet.
4. What percentage of the 100% speed is this? See if you can figure it out and write it in your worksheet.
5. Change the Roll step of the macro to roll at that speed and try it out. Watch the spot near the masking tape. Did Sphero stop rolling close to that point? Remember, it won't be exact because of the time it takes for Sphero to get up to speed.

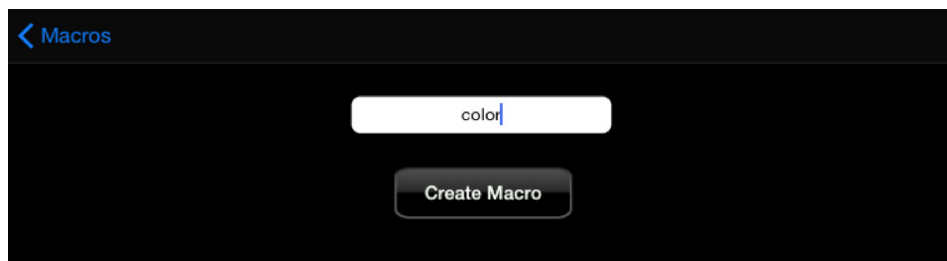
Part 5: Color

Let's switch now from movement to color, and you'll learn how to make Sphero light up with different colors. Every color that you see can be made up of a mixture of the colors red, green, and blue. Sphero lets you set the percentage of red, green, and blue that you use in this mixture. Follow these steps to play around with color percentages:

1. First, create a new macro for colors. Tap Macros on top to get to the list of macros. Then tap the + button at the bottom to create a new macro.



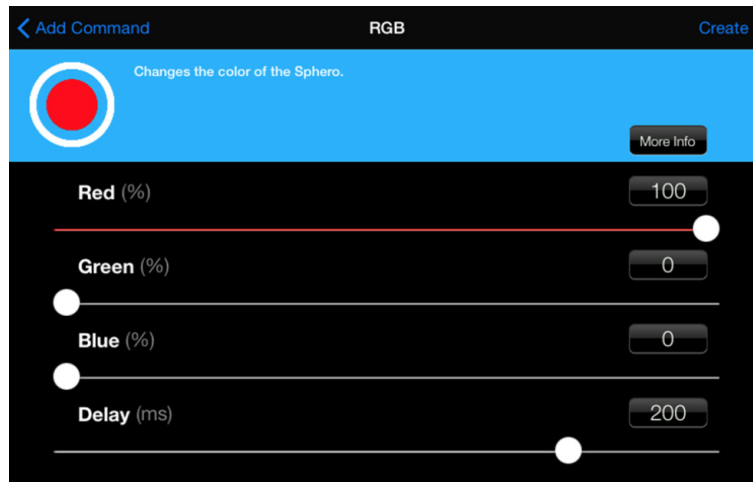
2. Name the macro color. Tap Create Macro.



3. Tap the Add (+) button at the bottom of the screen. Tap on RGB (which stands for "Red, Green, Blue"), the third command in the list.



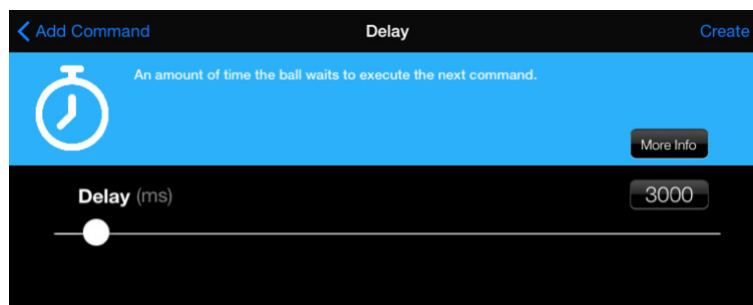
4. Drag the slider for Red to the right for 100% in order to make a red color. Slide the Green and Blue sliders to the left for 0%. Leave the Delay at 200. Click Create.



5. Now let's add a three second delay so that you can see the color for three seconds. Tap the + button to add a new step.
6. Tap the Delay button.



7. Set the Delay value at 3000 milliseconds (3 seconds). Click Create.



8. Set Sphero on a table. Tap Play and Sphero will turn red for 3 seconds.

That's the color we get with 100% red. Let's modify the color and make it 50% red. What do you expect it to look like?

1. Tap RGB and change the red value to 50%. Leave the others at zero.
2. Tap percent to get back to the macro list. Tap Play.

That's the color we get with 100% red. Let's modify the color and make it 50% red. What do you expect it to look like?

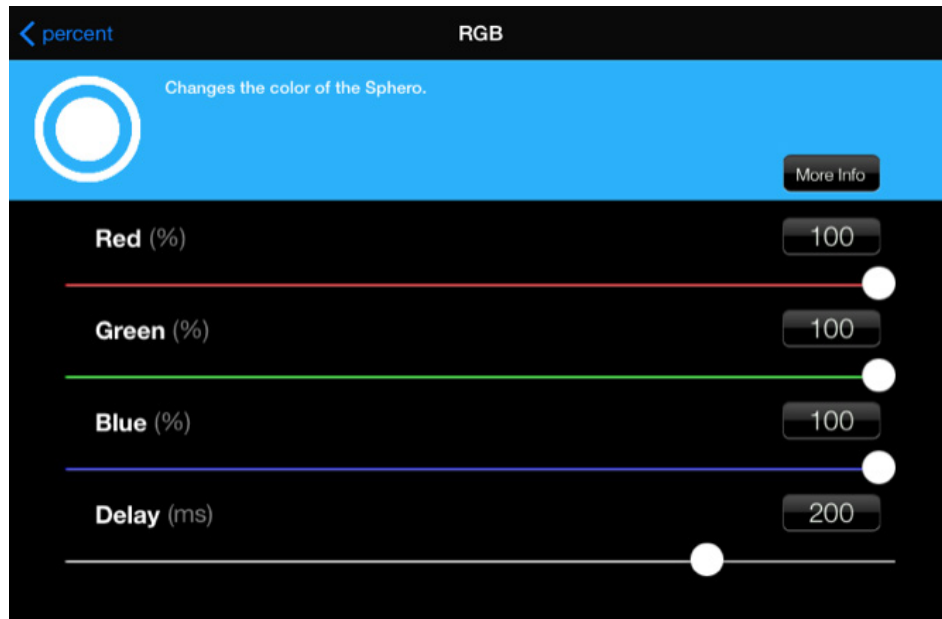
Now, let's make the color yellow. When you mix red and green paint, you get brown. But when you mix red and green light, you get yellow. Here's how:

1. Tap RGB and change the red and green values to 100%. Leave blue at zero.
2. Tap percent to get back to the macro list. Tap Play.

Let's say you wanted to make orange. Orange is like yellow, but it's got more of the red color in it. Try reducing the green to 50%, but leaving the red at 100% and the blue at 0%. Did that get orange?

Part 6: Color Challenge

For the challenge, let's try to make two new colors: gray and purple. To make gray, let's start by making white. White is the color you get when you have 100% red, 100% green, and 100% blue.



Try it out. Sphero won't be exactly white, but it will be pretty close.

What happens when you have 0% red, 0% green, and 0% blue? Try it.

As you might expect, Sphero puts out no light at all. No light at all is considered black.

You know how to make white and black, so how about gray? Gray is about halfway between white and black, so what is your guess? Once you figure out the percentage values for red, green, and blue, write them in your workbook.

Finally, once you have gray, try making purple. By now you have noticed the little circle at the top left when you are choosing your colors. You can use this to get to the color you want, even though it won't exactly match what's shown on Sphero. Play around with the sliders and see if you can get purple.

Fill in your color percentage values for purple on your worksheet.

MacroLab Lesson 4 Percentages: Teacher Guide

Overview

Students will use Sphero to explore percentages using speed and color. The speed experiments are quantitative and the color experiments are qualitative. They will program the Sphero to move at 100% speed for a particular amount of time, and then measure how far it has gone. Then they will repeat the experiment with 50% and 30% speeds. They will calculate the speeds based on distance and time, and then they will perform division to show that the smaller speeds are (roughly) the expected percentages of the 100% speed. Then they will be given a challenge to figure out what percentage of 100% speed is required to move Sphero to a certain position.

For the color experiments, students will make Sphero turn different colors. They will explore what happens when percentages less than 100% are used, for both single primary colors and mixtures of primary colors. For their challenge, they will make the Sphero turn gray and purple.

Read through the student guide. At the start of the lesson, go over the concepts of percentage and how the Sphero uses percentages.

Objective

Students will:

- Create a one-line program that moves the Sphero at a steady speed for a specified amount of time
- Perform measurements to determine the distance traveled.
- Perform division to calculate speeds
- Perform division to calculate percentages
- Create a two-line program to display a color for a certain length of time.
- Have Sphero light up in primary colors (red)
- Have Sphero light up in colors that involve two primary colors (yellow and orange).
- Have Sphero light up in colors that involve two or three primary colors (gray and purple).

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.4.MD.A.2: Use the four operations to solve word problems involving distances, intervals of time, etc.
- CCSS.MATH.CONTENT.5.OA.B.3: Analyze patterns and relationships
- CCSS.MATH.CONTENT.5.NBT.A.1: Recognize that in a multi-digit number, a digit in one place represents 10 times as much as it represents in the place to its right and $\frac{1}{10}$ of what it represents in the place to its left.

Common Core Math Standards (continued)

- CCSS.MATH.CONTENT.5.NBT.B.7: Add, subtract, multiply, and divide decimals to hundredths, using concrete models or drawings and strategies based on place value, properties of operations, and/or the relationship between addition and subtraction
- 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 iPad, but other devices could be used. Here is what each group would need:

- iPad with Sphero MacroLab loaded. You can get Sphero MacroLab for free from the iTunes app store.
- Sphero that has been fully charged
- Masking tape
- Tape measure
- Print-out of the worksheet (last page of teacher's guide)
- A flat clear path of at least 25 feet. (Preferably not very slippery.)

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: Speed

In Part 3, students will create a macro (computer program for the Sphero) and modify it in order to do an experiment. They will play the macro, measure how far the Sphero goes each time, record the values in on their worksheet, and calculate speed values and percentages.

Part 3: Speed (continued)

To create a macro to roll Sphero, they will:

1. Create a new macro
2. Add a Roll command
3. Set the Roll command to move at 100% speed for 3000 milliseconds (3 seconds).

Note that at 100% speed, the Sphero will go quite far in 3 seconds. Make sure you have room for it.

To run the experiment, they will:

1. Put a small piece of masking tape on the floor and put the Sphero on the tape.
2. Tap Play in MacroLab
3. Watch for where the ball stops rolling and starts sliding and place a piece of masking tape there. Then they will use the measuring tape to measure from the ball to the masking tape. They only need to be accurate within 1 inch or 2cm.
4. They will write the number down in the worksheet.
5. They will calculate the actual speed by dividing the distance by the time (3 seconds).
6. Then they will modify the macro so that the speed is 50% and they will repeat the experiment, recording the result in the worksheet, and calculating the speed.
7. Finally, they will modify the macro so that the speed is 30% and they will repeat the experiment, recording the result in the worksheet.
8. At the end they will divide the distance from the 50% experiment by the 100% experiment, and also the distance from the 30% experiment by the 100% experiment. They should have values close to 0.5 and 0.3, and they should be able to translate those into percentages.

Why won't their results be exactly 50% and 30%? The reason is that the Sphero (or any real object) takes time to speed up. This little bit of time at the beginning makes it so that it doesn't exactly travel the distance we would expect it to. They should especially be able to see how the Sphero takes time to speed up when running it at 100% speed.

Part 4: Speed Challenge

For part 4, the students will:

1. Place a piece of masking tape along the path of the Sphero, closer to where the 100% speed Sphero stopped rolling. (Preferably around 75% of the way.)
2. They will measure how far that distance is.
3. They will use their worksheets to calculate the speed percentage they would expect to have the Sphero stop rolling at that point.
4. They will run the macro and see how close they get.

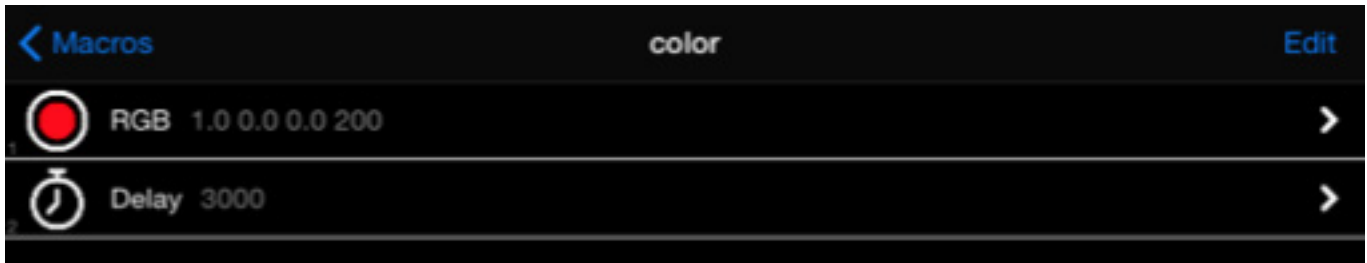
Part 5: Color

Students will experiment with mixing colors to light up the Sphero. Note that mixing colors with light is different than mixing colors with paint. When you mix red and green paint, you get brown; when you mix red and green light, you get yellow. Similarly, mixing all colors creates black for paint, but creates white for light. Note that the Sphero shell has some color, so that they colors that show up on the Sphero won't exactly match what's shown in MacroLab.

Part 5: Color (continued)

For part 5, the students will:

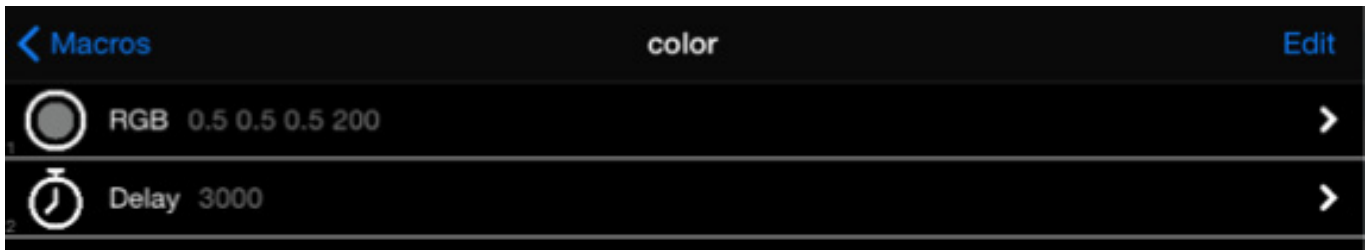
1. Create a new macro, adding a color (RGB) command and a delay command. The color will be set to 100% red and the delay will be set to 3000 msec (3 seconds).



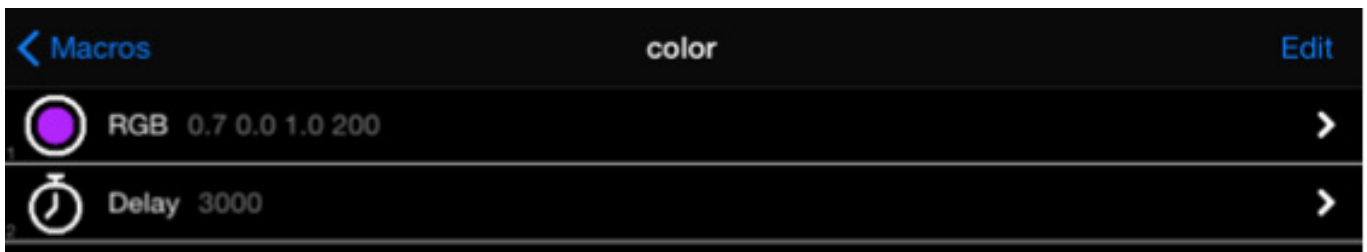
2. Modify the macro to see what happens if they only use 50% red. (It's a darker red.)
3. Modify the macro to make yellow with 100% red and 100% green.
4. Modify the macro to make orange (100% red, 50% green).

Part 6: Color Challenge

Students will experiment with creating two colors that use all three primary colors: red, green, and blue. First they will create white, which is 100% red, 100% green, and 100% blue. Then they are challenged to create gray, and hopefully they will realize equal parts of red, green, and blue, less than 100%, will create gray. (For example, 50% red, 50% green, and 50% blue.)



Lastly, they are challenged to create purple. This will take some playing around with percentages. In the end, they will hopefully end up with something like 70% red, 0% green, and 100% blue.



MacroLab Lesson 4 Percentages: Worksheet

Names:

Part 3 - Speed:

How far did the Sphero travel for 100%, 3 seconds: _____

What is the speed? (Distance / 3 seconds): _____

How far did the Sphero travel for 50%, 3 seconds: _____

What is the speed? (Distance / 3 seconds): _____

How far did the Sphero travel for 30%, 3 seconds: _____

What is the speed? (Distance / 3 seconds): _____

What is the 50% speed divided by the 100% speed: _____

- What is that in percentage: _____

What is the 30% speed divided by the 100% speed: _____

- What is that in percentage: _____

Part 4 - Speed Challenge:

How far is it between the starting and ending masking tape pieces: _____

What speed should it go for 3 seconds: _____

What percentage will you use for speed? _____

Part 6 - Color Challenge:

What percentages did you use to make gray? Red: _____ Green: _____ Blue: _____

What percentages did you use to make purple? Red: _____ Green: _____ Blue: _____

MacroLab Lesson 5

Patterns & Colors: Student Guide

You are going to be using Sphero to program patterns. Sphero can be programmed to change colors for different durations. You are going to be creating Macros that do just that. This lesson will challenge you to solve and generate patterns as well as send a message in Morse Code by programming your Sphero to light up in the correct series.

The MacroLab commands you are going to use in this lesson are:

 RGB – Changes the color of the Sphero.

 Delay – Waits for a while before doing the next step.

First you have to connect Sphero to the iPad (Part 1), and then there are three activities (Part 2 - Part 4), and finally a challenge to see if you can apply what you've learned (Part 5). An extra challenge using math patterns is provided in Part 6.

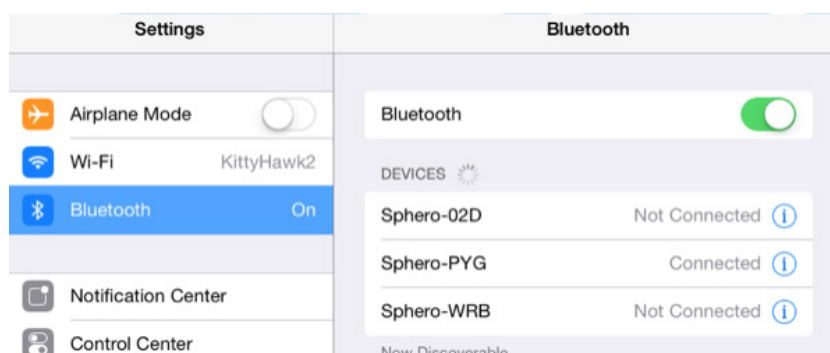
Colors and Patterns

For review, feel free to refer back to MacroLab Lesson 2 where you first explored programming Sphero to change colors. Patterns are all around us and it is important to be able to recognize and analyze as well as create patterns.

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".



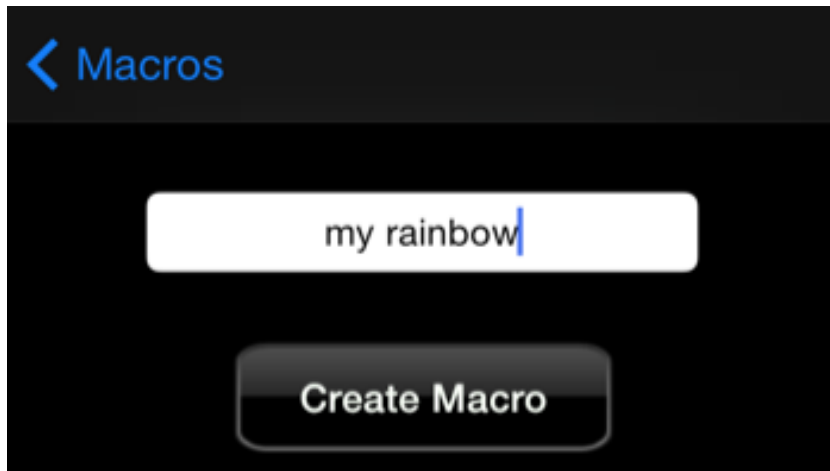
Part 2: Colors

Now that we have Sphero connected, follow these steps for the first experiment:

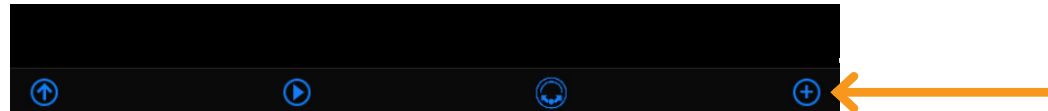
1. Select the preloaded program called Rainbows
2. Press play and watch the program
3. Click Macros in the top left corner to move back to the main MacroLab screen.
4. Now we are going to try to create a better rainbow
5. Tap the + button at the bottom to create a new macro.



6. Where it says Macro Name, type my rainbow. Click Create Macro.



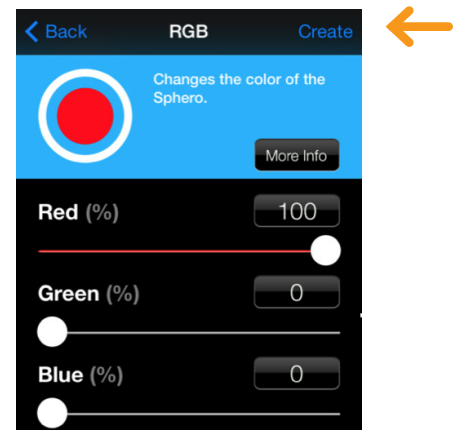
7. Add a command by tapping the + button at the bottom right.



8. Tap the Add (+) button at the bottom of the screen. Tap on RGB (which stands for "Red, Green, Blue"), the third command in the list.



9. Drag the slider for red to the right for 100% in order to make a red color. Slide the green and blue sliders to the left for 0%. Slide the delay to 0, we will use the next command to make the color delay for 1 second. Click Create.

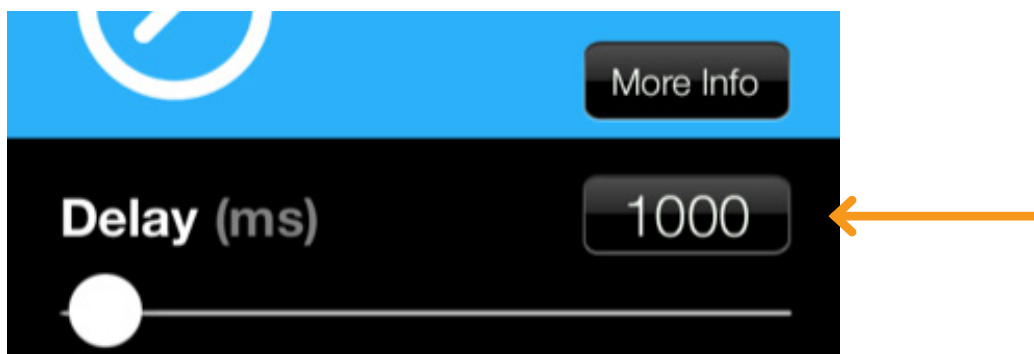


Part 2: Colors (continued)

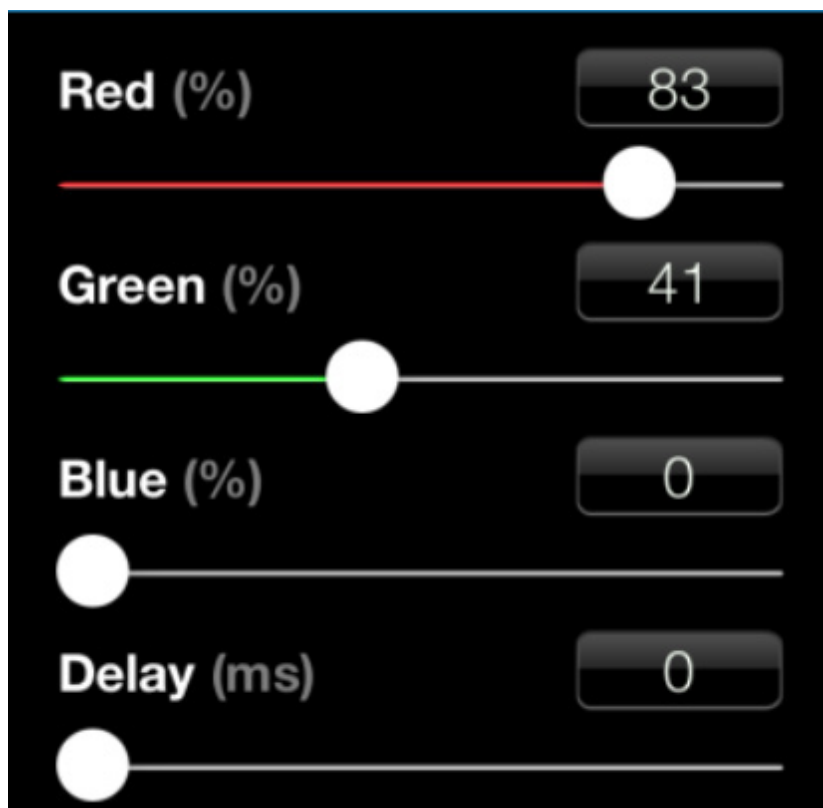
- Now let's add a one second delay so that you can see the color for one second. Tap the + button to add a new step.
- Tap the Delay button.



- Choose 1000 ms delay. It may be easier to click on the box and type in 1000 than using the slider. Click Create.



- Now you will create the next color. Tap the add (+) button, then tap on RGB. Drag the slider for red to the right to around 83% and drag the green slider to around 40% in order to make orange. The blue slider should be at 0%. Slide the delay to 0, we will use the next command to make the color delay for 1 second. Click Create.



Part 2: Colors (continued)

14. Tap the + button to add a new step.
15. Tap the Delay button.
16. Choose 1000 ms delay. Click Create
17. Repeat steps 8-12 above and look on the screen as you move the slider bars until you have created yellow.
18. Record on your worksheet what the slider bar values were to create yellow.
19. Repeat steps 8-12 for green.
20. Repeat steps 8-12 for blue.
21. Repeat steps 8-12 above and look on the screen as you move the slider bars until you have created purple.
22. Record on your worksheet what the slider bar values were to create purple.
23. Set Sphero on a table. Tap Play.

Do you see the rainbow?

Part 3: Pattern Solving

What colors are missing in the patterns?



Part 4: Make your own pattern for Sphero

1. Using the color key below, program your own unique pattern with Sphero. In the table a dot represents a short delay (200 ms) and a rectangle represents a longer delay (600 ms). Add periods of no light in between each shape to help recognize the pattern. Write your pattern on your worksheet.

*Note: when setting color delays, the max delay that you can set on the RGB screen is 255, so when programming 600 ms of a color, use a separate delay command after the RGB command.

An example of using the key to program:

A red dot would be coded as:

RGB->Red command with delay 200 ms



A red rectangle would be coded as:

RGB->Red command with delay 200 ms

Delay 400 ms



A break between colors would be coded as

RGB->No color command with delay 200 ms

2. Once you have built your Macro, see if another team member can solve your code and then try to solve theirs.

Part 5: Challenge

For our challenge, we will program Sphero to light up to send a message in Morse Code.

1. Use the table below to help figure out how to code the word your teacher gives you. The table says a dot is used for 1 unit, for us that will be 200 ms and a dash will be 3 units, or 600 ms.

Part 5: Challenge (continued)

2. Use a different color for each different letter
3. After you have coded your word let the teacher know
4. As a group try to figure out the entire secret message by going around to the different groups and figure out what each word is.

International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

A	● —	U	● ● —
B	— ● ● ●	V	● ● ● —
C	— ● — ●	W	● — —
D	— ● ●	X	— ● ● —
E	●	Y	— ● — —
F	● ● — ●	Z	— — ● ●
G	— — ●		
H	● ● ● ●		
I	● ●		
J	● — — —		
K	— ● —	1	● — — —
L	● — ● ●	2	● ● — —
M	— —	3	● ● ● — —
N	— ●	4	● ● ● ● —
O	— — —	5	● ● ● ● ●
P	● — — ●	6	— ● ● ● ●
Q	— — ● —	7	— — ● ● ●
R	● — ●	8	— — — ● ●
S	● ● ●	9	— — — — ●
T	—	0	— — — — —

Part 6: Optional Extra Math Pattern Challenge

If we assign each number to a shape, as in the table below, we can make mathematical patterns.

1	=		7	=	
2	=		8	=	
3	=		9	=	
4	=		10	=	
5	=		11	=	
6	=		12	=	

Use the Key above to figure out the missing colors in the patterns below:

1. The first couple numbers are given for this first pattern. What are the missing shapes? What is the mathematical pattern?



2. What are the missing shapes? What is the mathematical pattern?



MacroLab Lesson 5

Patterns & Colors: Teacher Guide

Overview

In this lesson students will work with patterns, using Sphero as they analyze, generate and solve them. They will also explore how to make Sphero turn different colors for different amounts of time. At the end of the lesson there is a challenge to see if the students can use Sphero within their small groups to send a message in Morse code and then decode the messages of the whole class. It may be useful to discuss Morse code and how it has been used in history. Read through the student guide.

Objective

Students will:

- Create a multi-line program to display many colors sequentially, each for a certain length of time.
- Have Sphero light up in primary colors (red)
- Have Sphero light up in colors that involve two primary colors (yellow and orange)
- Have Sphero light up in colors that involve two or three primary colors (purple).
- Program Sphero to light up with colors in an original pattern
- Solve patterns based on colors
- Send a Morse code message using Sphero

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 Macrolab loaded. You can get Sphero Macrolab for free from the iTunes app store.
- Sphero that has been fully charged
- Print-out of the worksheet (last page of teacher's guide)

Part 1: Connect the Sphero

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

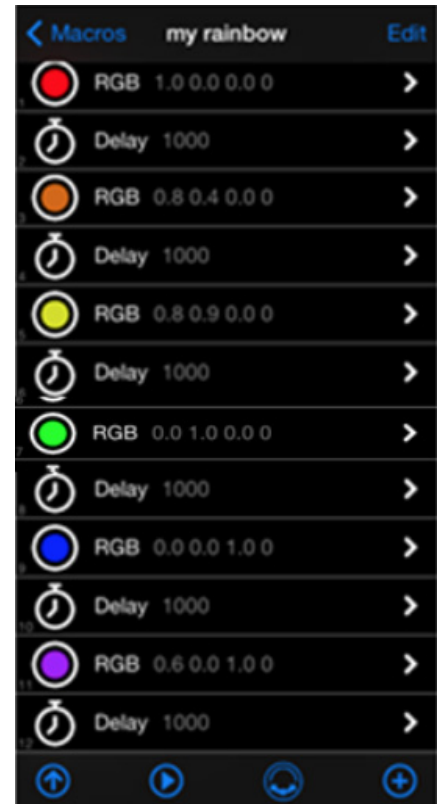
1. Wake up the Sphero
2. Turn on Bluetooth
3. Connect the correct Sphero to the iPad, using the colors flashes as a way to tell which Sphero has which name.

Part 2: Code the Rainbow

To ensure students know how to program each color that will be used later in the worksheet they begin by programming a rainbow with each color showing for 1 second. There is a preloaded macro that is called rainbow but this part challenges students to make a better rainbow than the preloaded macro.



The final macro (right) should look something like this:



Part 3: Pattern Solving

Part 3 does not involve Sphero, instead students will solve the color patterns provided in the student guide. The solutions (below) are:

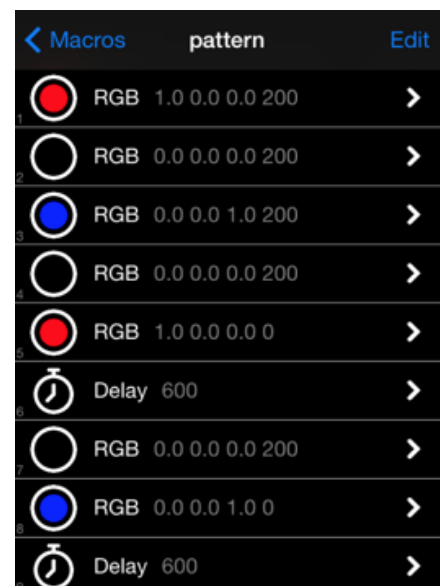
- 1.
- 2.

Part 4: Pattern Generating

Using a key with colors and shapes that is in the student guide (and you can see it to the left), the students will develop their own patterns. Each color and shape correspond with coding commands and the students will create macros utilizing this key. The students will then solve each other's patterns. There are examples in the student guide of how to get started.

An example macro (right) that corresponds with the pattern:

Red Circle – Blue Circle – Red Square – Blue Square



Part 5: Challenge

You will provide each group with a word which is part of a message. Using a provided Morse Code table the students will make Sphero light up the word you gave them. To make it easier for decoding, it is recommended to the students that each letter should light up a different color.

Once all of the groups have coded their Sphero, have them play it back for the whole class and see if the students can decode each of the words that the other groups programmed. Once all words have been decoded, they can be put together to form the secret message. It may be helpful to turn off the lights to help see the colors better.

Two potential secret code sentences that could be used are:

- Sphero makes coding fun
- Coding sphero colors rocks

Part 6: Optional Challenge 2

See the color-number key in the student guide.

1. 12 9 10 7

___    ___    ___ ___ 

Missing colors:    

Pattern: 12,9,10,7,8,5,6,3,4,1,2

The mathematical pattern for the first question is: -3, +1, -3, +1

2.

     ___  ___   ___ 

Missing colors:   

Pattern: 12,6,10,5,8,4,6,3,4,2,2,1

The mathematical pattern for the second question is: divide by 2, add 4, divide by 2, add 3, etc.

OrbBasic Lesson 1

Goto and Variables: 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

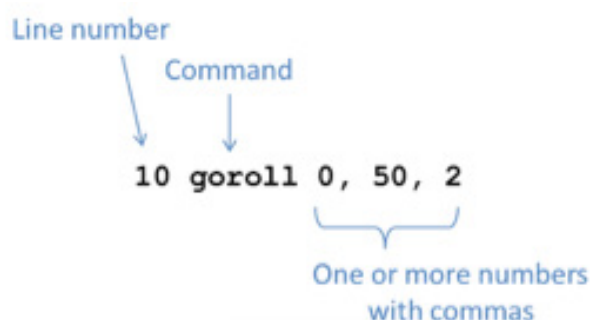
First you have to connect the Sphero to the iPad (Part 1), then aim your Sphero in the correct direction (Part 2). In part 3, you'll write your first OrbBasic program to make Sphero roll out and back, then you'll make it repeat itself (Part 4), and then you'll control the distance it rolls with a variable (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 look 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 numbered 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, it will be difficult. 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.

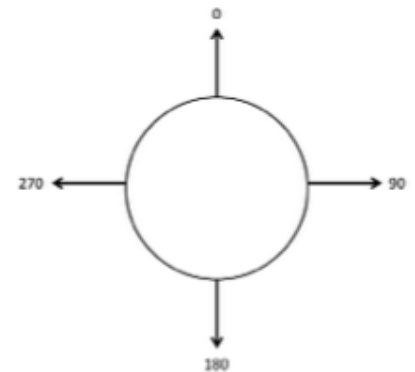
The second piece of text is the command. This will tell Sphero to do a particular action.

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)
- 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 goroll 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
```

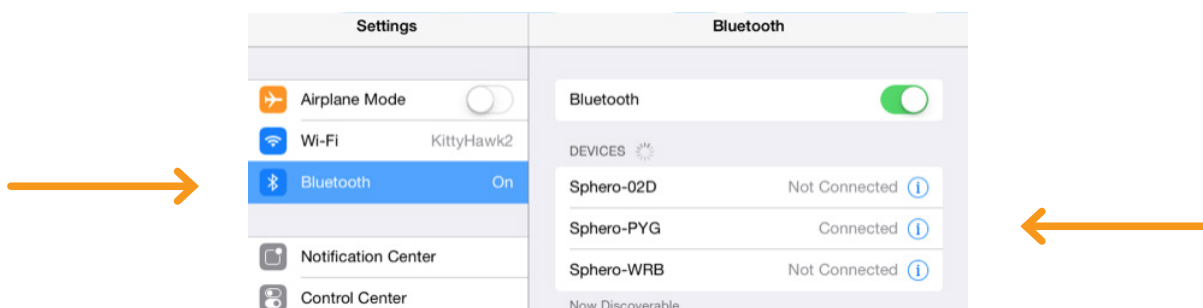
means to take what's stored in the b variable, add 5, and then store the new number in the b variable. If b were 35, then after doing this line, it would be 40.

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

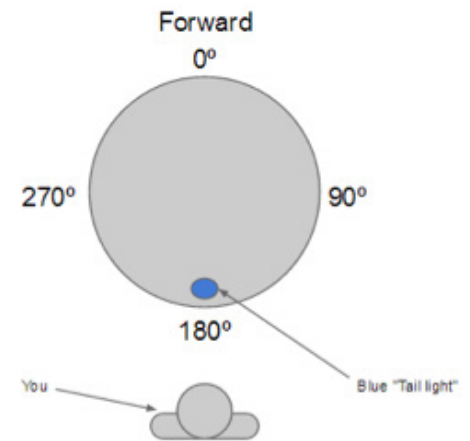
First thing you need to do is to connect the iPad to Sphero. Here's how:

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!

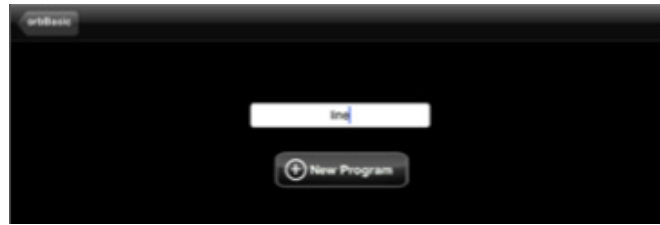
Part 3: Your first OrbBasic program

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.



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



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



- 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
```

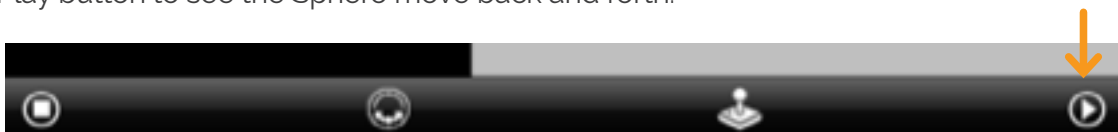
- Tap the Done button in the upper right hand corner.



- 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
```

8. Now we should see the ball rolling back and forth. Tap the Play button to see it work.
9. 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!

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:

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 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:

- Wake up the Sphero
- Turn on Bluetooth
- 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.

```
Generating Fragments...
Generated 1 fragments
Sending program to robot...
Executing Program...
```

```
Line 30: Bad numeric parameter
```

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: with lines spaced by 10, so it's easy to squeeze new ones in.

```
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
```

OrbBasic Lesson 1 Goto and Variables: Worksheet

Names:

Part 5 - Challenge:

Write down your program that starts at 5 seconds back and forth and then drops by 1 second each time the Sphero goes out and back.

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.

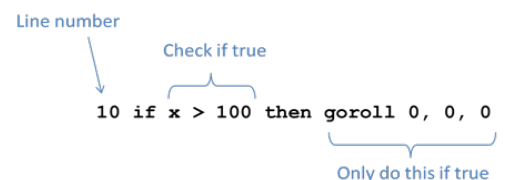


Next, we increase it to 10 degrees and roll a little more. Then 15 degrees. Then 20 degrees. We add 5 degrees to the heading each time.

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

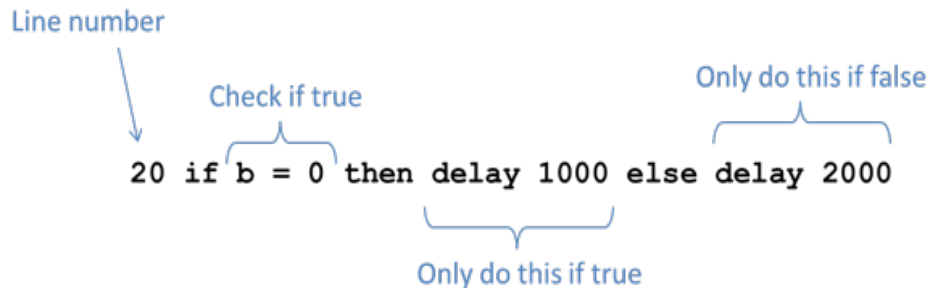
If/then and if/then/else statements

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:



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:



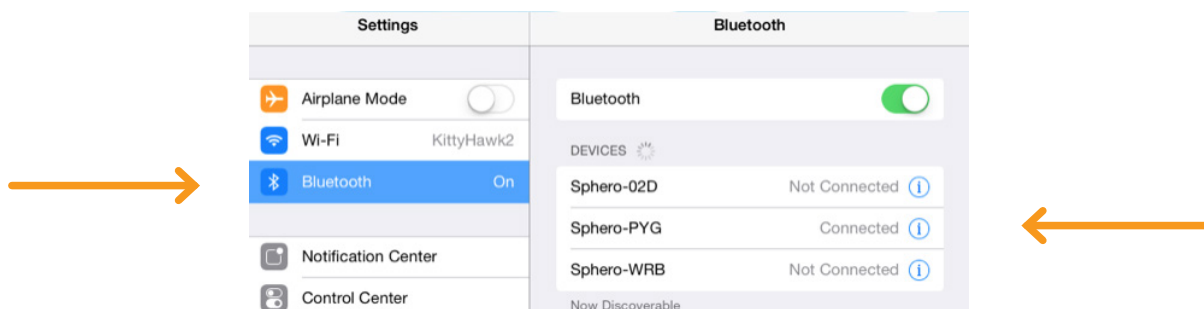
This line is saying that if the variable b is 0, then the Sphero should delay for 1000 milliseconds (1 second), but if b is not 0, then it should delay for 2000 milliseconds (2 seconds).

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:

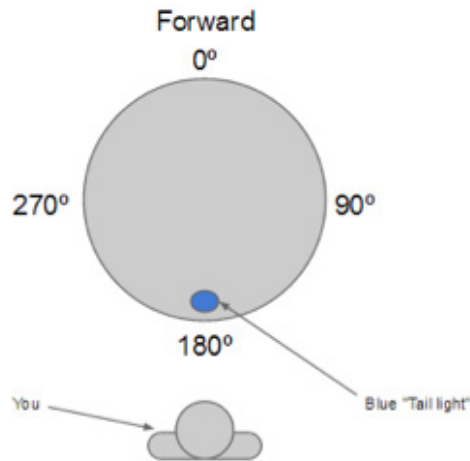
1. Pick up Sphero from its charging station and tap it twice on the logo to wake it up. You may have to tap it hard. It will start flashing colors when it is awakened out of its "sleep" state.
2. On your device, make sure Bluetooth is enabled. From the home page, click on Settings at the bottom. Then 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)

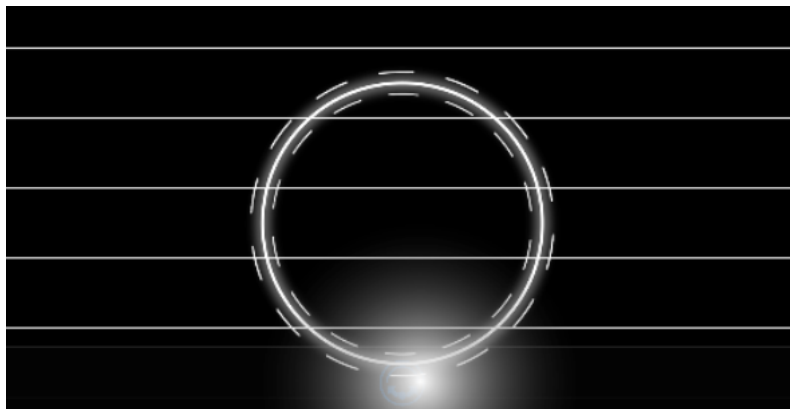


Follow these steps to aim the Sphero:

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

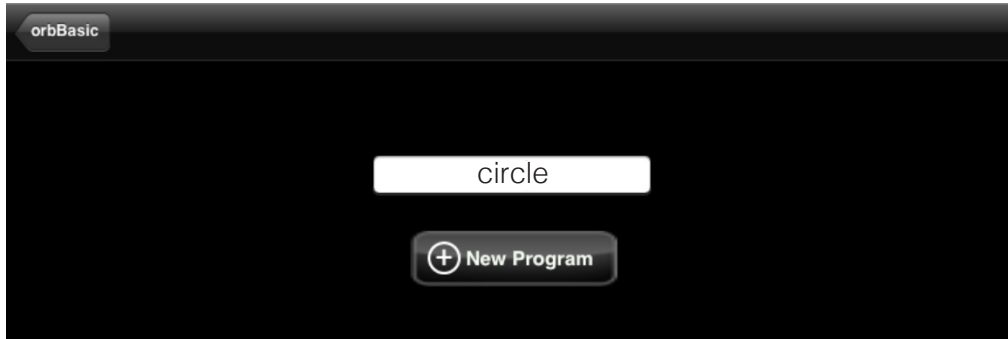
Now that we have the Sphero going in the right direction, let's get it going in a circle. As explained above, we want to increase the heading by 5 degrees each time. We'll need a variable called h (for heading) that starts at 0, and then we'll add 5 to it after each small roll. Follow these steps:

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!

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:

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 (last page of teacher's guide)
- A flat clear area of at least 10 x 10 feet. (Preferably not very slippery.)

Part 1: Connect the Sphero

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

- Wake up the Sphero
- Turn on Bluetooth
- 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.

Part 4: Adding Color (continued)

Although you could use OrbBasic's RGB 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: Challenges

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
```

OrbBasic Lesson 2
Circles and if/then/else: Worksheet

Names:

Part 4 - Challenge:

Write down your program that draws circles that show two different colors and that has a bigger circle each time it goes around.

OrbBasic Lesson 3

Sensors and Random: Student Guide

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.

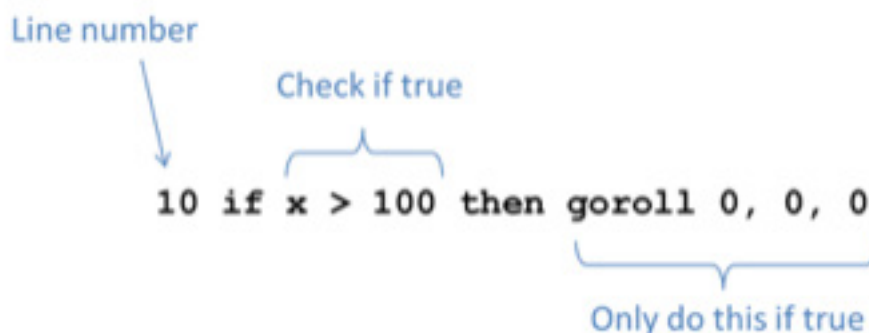
Here are the Sphero commands you'll be using for this lesson:

- 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:



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:

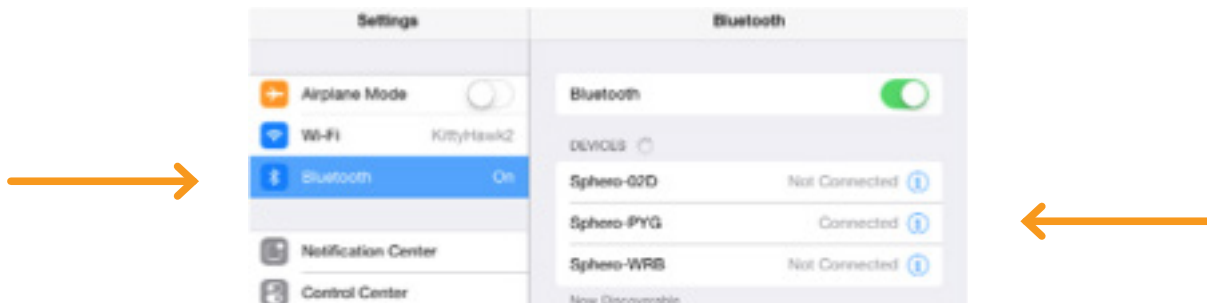
```
50 LEDC 4
```

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

Part 1: Connect the Sphero

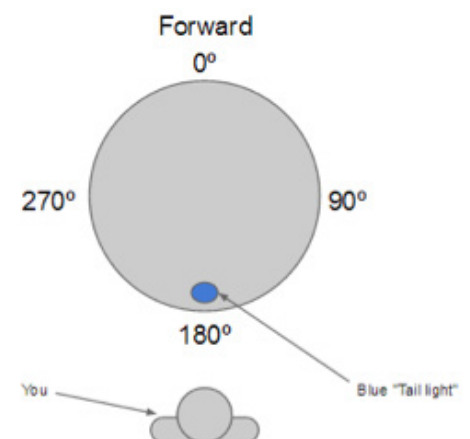
First thing you need to do is to connect the iPad to Sphero. Here's how:

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.



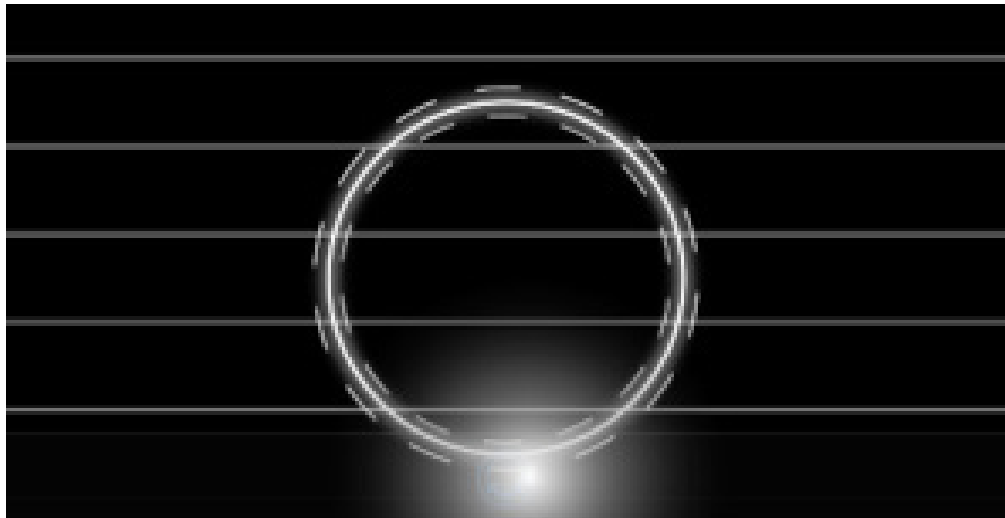
Part 2: Aiming Sphero (continued)

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.



Part 3: Making Sphero turn purple in the air

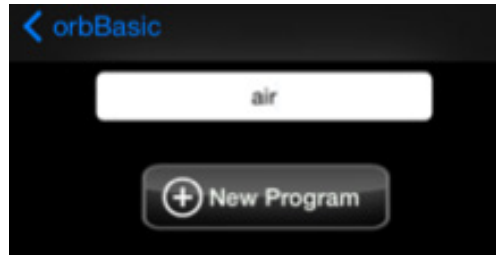
Sphero has an accelerometer inside, an accelerometer is used to measure the acceleration, or change in velocity of Sphero.

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.

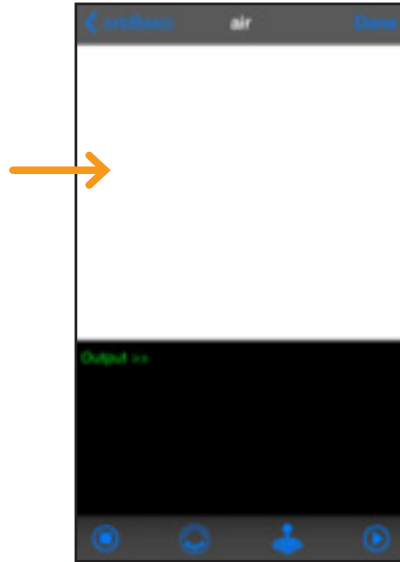
1. Tap the + button at the bottom to create a new program.



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
```

- Using accelone, detect if a collision happens  
`20 if accelone > 5000 then goto 30 else goto 10`
- Now you need to set Sphero to turn red using the command below for the case where the conditional statement is true  
`30 LEDC 1`
- 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 delay 1000`
- Finally the program needs to jump back to beginning  
`50 goto 10`
- 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.

- 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.
- You may want to set an initial color for Sphero as well.
- Next have Sphero to roll in a straight line
- 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`)
- Don't forget to add a delay after this line to keep sphero moving in that direction for a half a second.
- 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!

## OrbBasic Lesson 3

### Sensors and Random: Teacher Guide

#### 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:

- Wake up the Sphero
- Turn on Bluetooth
- 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):

```
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
```



## OrbBasic Lesson 3 Sensors and Random: Worksheet

### 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:

## Overview

SPRK STEM challenges are fun, interactive activities that challenge students to use creativity and teamwork to move through simple steps of the design process in order to build Sphero-based creations. In this challenge, there are optional activities that provide students time to learn about chariots as they were used throughout history. Students will brainstorm designs for their chariots both individually and in groups. The teams will then have sufficient time to build their own Sphero chariot to race. Also, as a class the students build a macro for Sphero to navigate the course.

## Getting started

To run this challenge, you will need Spheros. 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 or other tablet devices, but other devices could be used. Here is what each group would need:

- iPad or tablet with Sphero Macrolab loaded. You can get Sphero Macrolab for free from the iTunes app store or Google Play.
- Sphero that has been fully charged

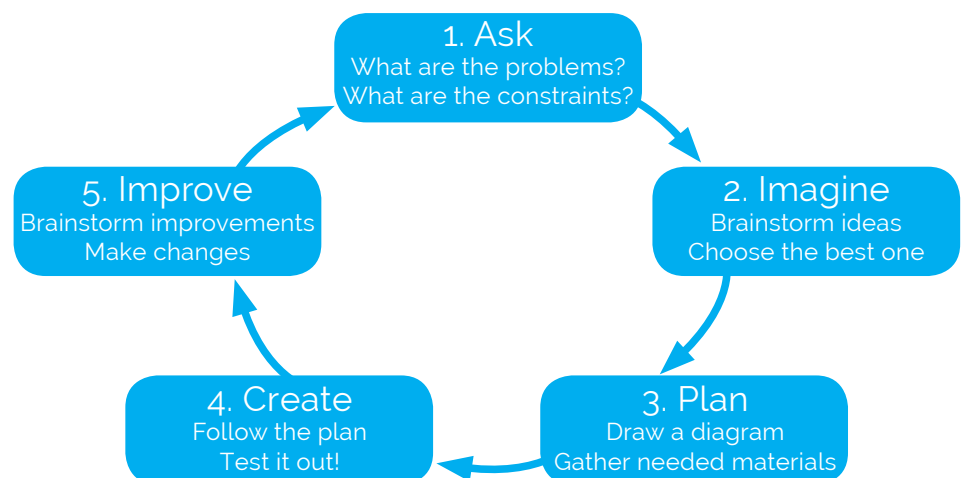
## Materials

For this challenge, the following materials are recommended. Feel free to be creative and use different materials for your class.

- Materials for Chassis and Wheels (Legos, Knex, cardboard, CDs, etc.)
- Large space on the floor for building the race track
- Tape to outline the course

## PART 1: Introduction – 15 minutes

- Break students into groups
- Briefly introduce Sphero and how it works
- Describe engineering design process (right)
- Describe the Chariot Challenge and show students the different materials available to build with, as well as, any guidelines they need to follow



## PART 2: Individual brainstorming – 5 minutes

This section gives students time to generate concepts on their own, it may be challenging to think of 8 unique ideas but challenge them to think of crazy and weird designs.

- Each student has a blank piece of paper and folds it into 4 quadrants. Using both sides of the page have them draw 8 different potential designs

### Optional Social Studies focused activities (Steps 3 – 5)

## PART 3: Historical research in groups - 60 minutes

Students will perform research on chariots throughout history; the primary focus of their research will be around the design and function of them

- Assign each group to a historical time/region
- Provide the students with some information about the chariots in that culture or they just do web searches.

## PART 4: Historical presentations - 30 minutes

Have each group come up and present for 4-5 minutes. Each presentation should include the following:

- Photo of a chariot from their region
- What material were they made of?
- What were chariots used for?
- How many wheels and how big were the wheels?
- How many horses/other animals were used to pull them?
- 1 other interesting fact about chariots in that culture

While each group presents, you can fill in a table that is drawn on the whiteboard that describes the different characteristics of the different chariots.

## PART 5: Individual re-brainstorm - 10 minutes

After learning about the different chariots from history, have students fold another piece of paper into quadrants and sketch 8 new ideas. Each student identifies their favorite designs to bring to the team.

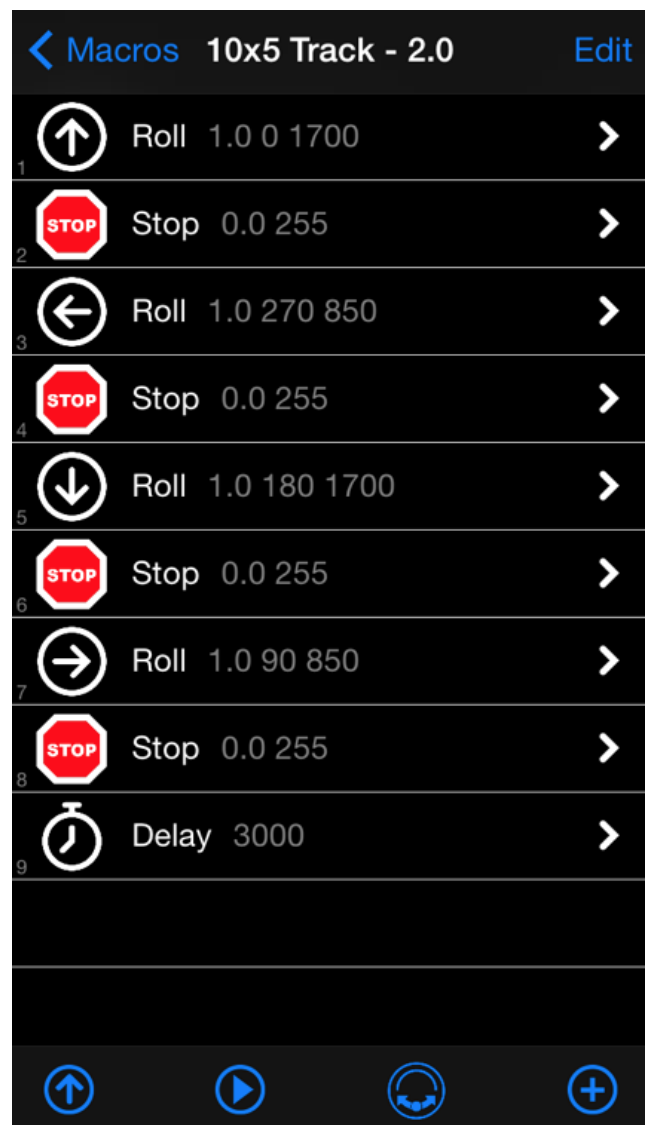
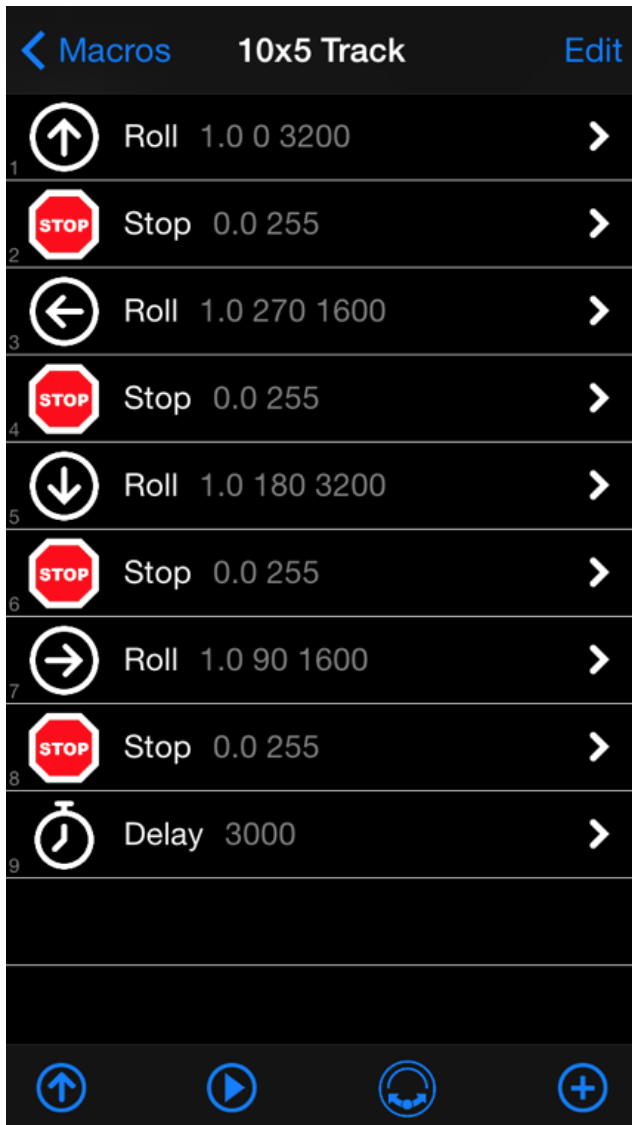
## PART 6: Build the track as a class – 30 minutes

Build simple track on the floor with masking tape.

## PART 7: Create the chariot Macro - 30 minutes

The screenshots on the following page display the macros for a track 10 feet X 5 feet. With a little guess and check, you can easily modify the Macro to fit dimensions of the course your class has built.

## PART 7: Create the chariot Macro (continued)



## PART 8: Groups pick design they will build – 20 minutes

As a group, students will choose a design to build and what materials they plan to use.

## PART 9: Build and test chariot – 90 - 120 minutes

Groups have time to build and test their chariots

## PART 10: Present Chariot and why you think it will work - 20 minutes

Each group will present for 3-4 minutes about the following:

- Why they chose the design they did
- What the hardest part of building was
- How they expect it to perform, do they anticipate any problems?

## PART 11: Race chariots against autopilot - 60 minutes

Teams race chariots against the autopilot Sphero in tournament or other format.

## PART 12: Reflection and discussion – 40 minutes

Each individual should write up their reflections on the activity and discuss their findings as a class. Some potential prompts and questions that you may want to ask are:

- What worked and what didn't?
- How would each student do things differently in the future?
- Why do the students think that the culture they studied used the chariot that they did?
- What materials worked best?
- What was the most challenging part of the activity?
- How did the size of the wheels or other design characteristics impact the results?
- What materials worked best?
- What was challenging and what worked well within your team?

## PART 13: Share your experience on @SpheroEdu

We love seeing SPRK in action! Tweet us a few photos and we'll share them with the world!

PLAY IS A  
POWERFUL  
TEACHER

## Overview

SPRK STEM challenges are fun, interactive activities that challenge students to use creativity and teamwork to move through simple steps of the design process in order to build Sphero-based creations. In this challenge students will build bridges from finite resources and then program Sphero to drive across. Students have the opportunity to research bridges and brainstorm potential concepts, as well as determine design specifications, and effectively communicate their vision for the bridge before building it.

## Getting started

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 or other tablets, but other devices could be used. Here is what each group would need

- iPad or tablet with Sphero Macrolab loaded. You can get Sphero Macrolab for free from the iTunes app store or Google Play
- Sphero that has been fully charged

## Materials

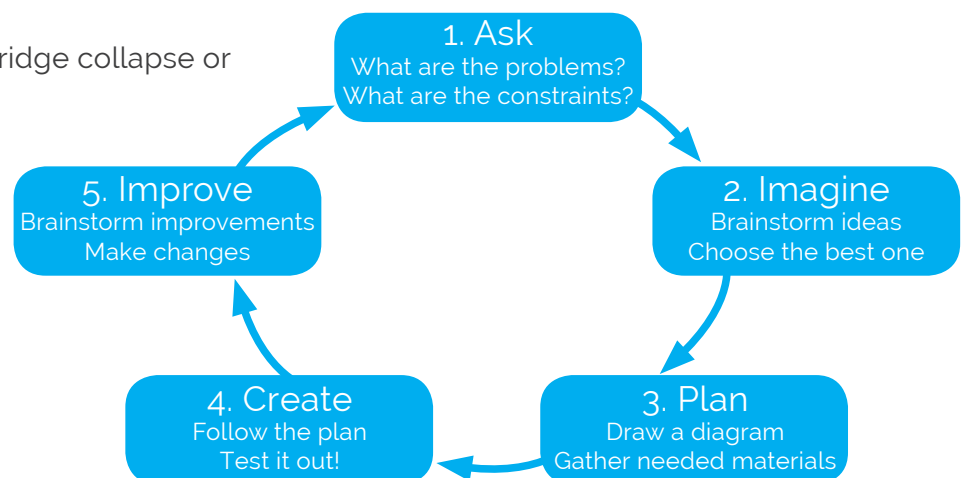
- Popsicle sticks, toothpicks, uncooked pasta, balsa wood, cardboard or other building material
- Glue, tape
- String
- Measuring tape or rulers
- 2 tables or other objects to span the bridge across

## PART 1: Introduction – 20 minutes

Gather materials for students to build bridges out of and set up the gap that students will build bridges to span.

- Break students into groups
- Watch video of Tacoma Narrows Bridge collapse or another famous bridge collapse.
- Briefly introduce Sphero to class
- Outline the engineering design process (right)
- Pose the challenge to the class:

**Sphero can drive on the table but it needs to get across the gap between 2 tables (or other objects). (We have found 12-14 inches is a good bridge distance to span)**



## PART 1: Introduction (continued)

- Each group has a set of materials (Popsicle sticks/pasta/balsawood/toothpicks/cardboard).
- You can provide each team with a mystery material that they have to incorporate into their bridge for an added challenge
- You can also include different incentives to encourage students to design efficiently like a prize for the lightest successful bridge.

## PART 2: Specifications – 20 minutes

In this part of the challenge students should gather information so that they can create a list of design specifications. These can include physical dimensions or specifications regarding material limitations or time constraints. Students can also set goals for the weight of their bridge.

- Have students take measurements of Sphero
- Students also will need to measure size of gap they need to span
- Students should determine what the road surface needs to be made of so Sphero can drive across it
- Have students create a list of 5 or more specifications for their bridges

## Optional physics/science focused activities (Steps 3 – 4)

### PART 3: Bridge research in groups - 60 minutes

The teams will research different types of bridges and then use this information to develop their designs. Groups are each assigned to different types of bridges. Some bridge types that could be assigned include: Arch, Beam, Cable-Stayed, Suspension, and Truss. You may want to just assign subtypes of a certain type, such as different types of Truss bridges.

### PART 4: Presentations - 30 minutes

Each group has 4 minutes to show a picture of the type of bridge that they researched as well as primary advantages/disadvantages and any notable features.

## PART 5: Brainstorm – 45 minutes

Provide teams with the following guidelines for brainstorming; as developed by the Stanford D School:

- One conversation at a time.
- Go for quantity
- Headline! (Quick, Concise Ideas)
- Build on the Ideas of others
- Be Visual
- Encourage wild ideas
- Stay on topic
- Defer Judgment

Set a goal for each team to come up with 20 different ideas in 20 minutes. Once teams have their ideas have them determine as a group which idea they would like to use

## PART 6: Drawings and materials – 30 minutes

In Part 6, teams create a clear drawing of their bridge and determine what materials they will need.

- As an option, ask students for dimensioned drawings. Each group, or each student should create a drawing of the bridge with the correct dimensions labeled on it.
- As an option, ask students to create a Bill of Materials. Each group should create a table or list of the materials they will need to build their bridge as well as the amount of each material.

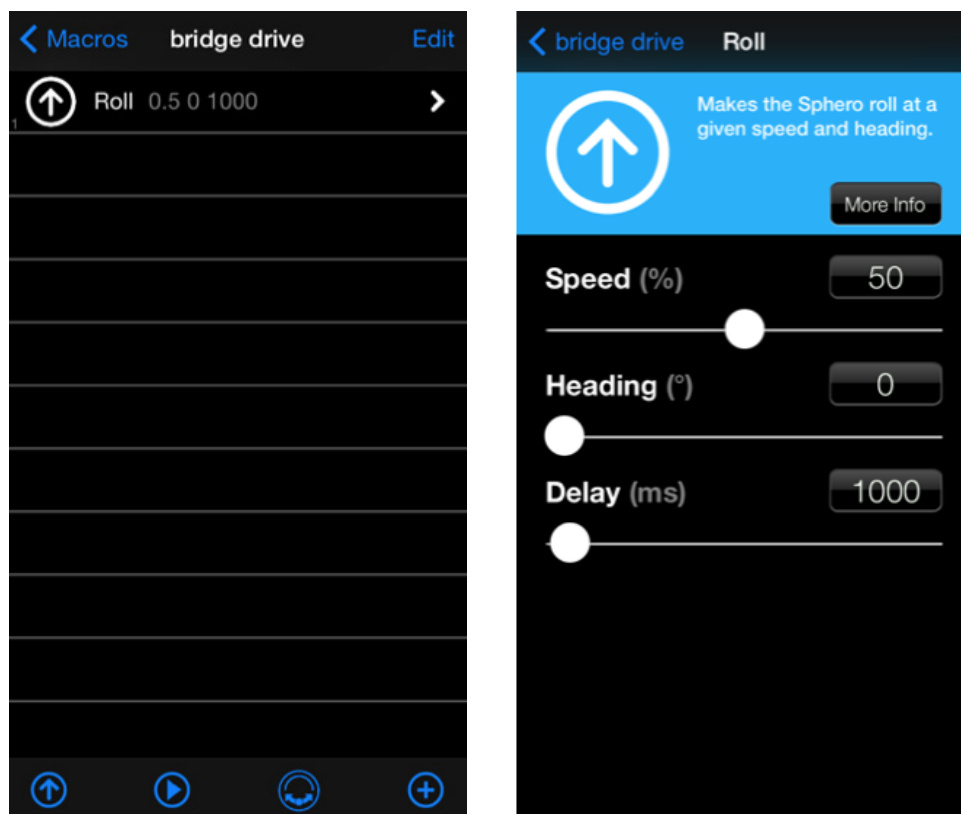
## PART 7: Present – 30 minutes

Each team should make a short presentation (4-5 minutes) about their bridge plan. If the optional section above (Dimensioned Drawing and Bill of Materials) was completed have the students present those sections as well. Each group should be able to present about the following.

- Why they believe their bridge will be successful
- What materials they will use
- What they anticipate will be the hardest part of the building process
- Any notable features of their bridge

## PART 8: Build bridges and program Sphero – 2 - 3 hours

- Teams build their bridges and if they finish early they can test them. Each team will also need to write a simple macro that gets Sphero across their bridge; it can be only one line long. See an example below:



## PART 9: Bridge Testing – 60 minutes

Finally it is time to test the bridges! If you offered incentives building the lightest bridge, you may want to begin by weighing all of the bridges.

- Each group should set up bridges across the gap.
- Next, each group should try out their macros to see if they can successfully get Sphero across the bridge.
- Each group should record whether or not their bridge was successful and if it failed note where the failure occurred.



## PART 10: Reflection and discussion– 45 minutes

Each individual should write up their reflections on the activity. Some potential prompts and questions that you may want to ask are:

- What worked and what didn't?
- How would each student do things differently in the future?
- What materials worked best?
- What bridge type worked best?
- What was the hardest or most fun part of the challenge?

## PART 11: Share your experience on @SpheroEdu

We love seeing SPRK in action! Tweet us a few photos and we'll share them with the world!

PLAY IS A  
POWERFUL  
TEACHER

## Sphero Hydro-Hypothesis STEM Challenge

Total time: 3.5 Hours  
Setting: Outdoors

### Overview

SPRK STEM challenges are fun, interactive activities that challenge students to use creativity and teamwork to move through simple steps of the design process in order to build Sphero-based creations. The Hydro-hypothesis challenge leverages Sphero's ability to swim to test whether or not students can build a contraption to transport a load (pennies) across a small body of water. This activity would provide an excellent opportunity to discuss the concepts of buoyancy, density, surface area, and what types of materials float in water.

### Getting started

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:

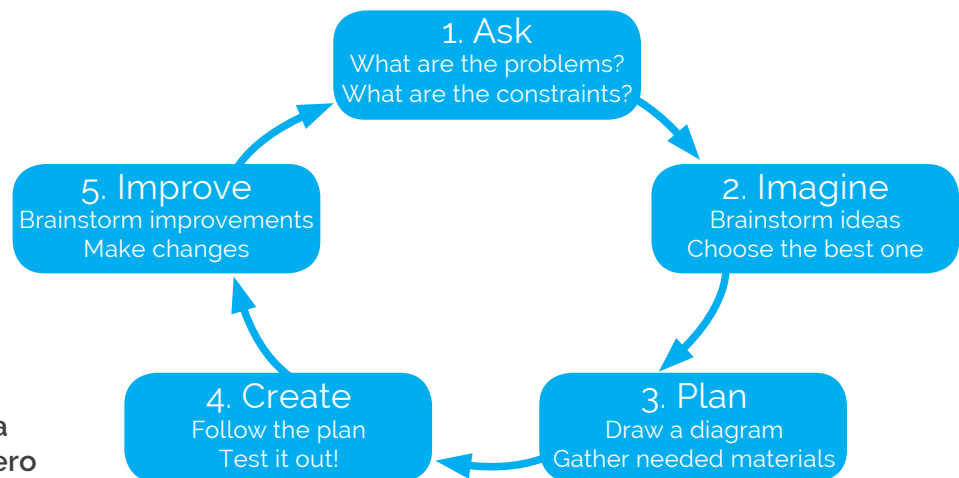
- iPad or tablet with Sphero Macrolab loaded. You can get Sphero Macrolab for free from the iTunes app store or Google Play
- Sphero that has been fully charged

### Materials

- Small swim pool or large cooler to fill with water
- Pennies to test how much the rafts can hold
- Materials to build Sphero watercrafts (foam, cardboard, rubber bands, string, Duct tape, any other materials that float)
- Scissors

### PART 1: Introduction – 15 minutes

- Break students into groups and briefly introduce Sphero
- Explain that for this challenge they will use the engineering design process.
- Show them Sphero swimming in water; have them take note of how much of Sphero is under water – this will help later when completing the hydro-hypothesis challenge!
- Pose the challenge: **Can you build a watercraft that is powered by Sphero using the materials we have here? It must be able to reach the other side of the pool while holding a load of 10-20 pennies.**



## **PART 1: Introduction (continued)**

- Show the students what materials are available
- As a class, build a list of some specifications that are required for a watercraft to be successful in the challenge

## **PART 2: Individual brainstorming – 10 minutes**

This section gives students time to generate concepts on their own, it may be challenging to think of 8 unique ideas but challenge them to think of crazy and weird designs..

- Each student has a blank piece of paper and folds it into 4 quadrants. Using both sides of the page have them draw 8 different potential designs
- Each student should then pick their favorite design to bring to the team

## **PART 3: Group brainstorm and selection of top design – 20 minutes**

- Teams review each member's ideas and see if they can generate any new ideas
- Once they have their ideas generated, each team determines as a group which design and materials they would like to use

## **PART 4: Present – 20 minutes**

Each team presents a picture of their selected idea and a quick description of why they believe it will be successful. You can also have them explain what they believe will be the most challenging part of construction.

## **PART 5: Build – 60 minutes**

Students have time to build their watercrafts; it is recommended that they test frequently in the water to make rapid changes. It can be tricky to build a Sphero-powered watercraft because Sphero is partially underwater when it swims.

## **PART 6: Final test – 30 minutes**

Time to find out which creations will sink and which will swim!

## **PART 7: Reflection and discussion – 45 minutes**

Each individual should write up their reflections on the activity. After the individual reflection, have some time for open discussion. Some potential prompts and questions that you may want to ask are:

- What worked and what didn't?
- How would each student do things differently in the future?
- What happened the first time they tested their watercraft?
- How did their watercraft change from the initial design?
- What materials worked best?
- What happened when more weight (pennies) were added?
- What was the hardest or most fun part of the challenge?

## **PART 8: Share your experience on @SpheroEdu**

We love seeing SPRK in action! Tweet us a few photos and we'll share them with the world!

Example of a Sphero Hydro-hypothesis contraption:



## Sphero Maze Mayhem STEM Challenge

Total time: 4.5 Hours  
Setting: Classroom

### Overview

SPRK STEM challenges are fun, interactive activities that challenge students to use creativity and teamwork to move through simple steps of the design process in order to build Sphero-based creations. In this challenge students will program Sphero to navigate an original maze built by you or by the class. This challenge will require students to gather data about the best route through the maze and figure out how to build a program so that Sphero can successfully navigate through the mayhem.

### Getting started

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 or other tablets, but other devices could be used. Here is what each group would need:

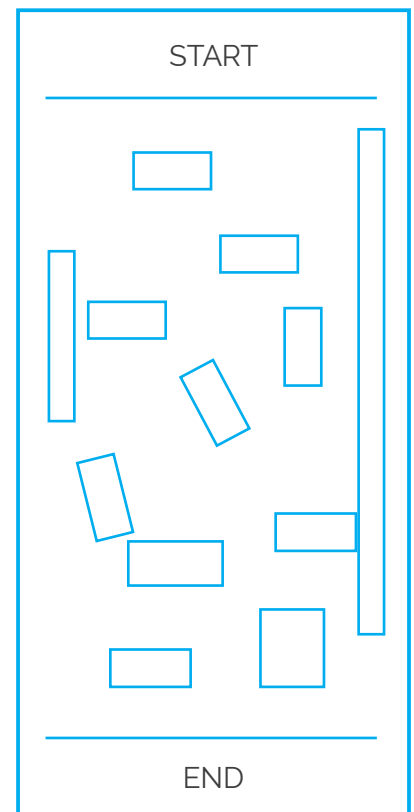
- iPad or tablet with Sphero Macrolab loaded. You can get Sphero Macrolab for free from the iTunes app store or Google Play.
- Sphero that has been fully charged

### Materials

- Books and other objects to build the maze
- Measuring tape or rulers
- Protractors
- Large space on the floor for building the maze
- Stopwatch
- Masking tape

### PART 1: Introduction – 15 minutes

- Break students into groups
- Briefly introduce Sphero and how it works
- Describe the Maze Challenge
- This STEM challenge is unique because the primary activity will be programming rather than building
- We suggest students take on specific roles for Parts 3 and 5 (Data Gathering and Programming). A couple of these roles could be:
  - Part 3: measurer, recorder, drafter/drawer, calculator
  - Part 5: reader (reads instructions written in part 3 to programmer), translator, programmer, & tester
- In order to ensure all students get exposure to the different roles it may be useful to have them switch roles every 15-20 minutes.



## PART 2: Maze Building - 15 minutes

Using books and other objects create a maze for Sphero to navigate, add interesting obstacles (such as water features) and narrow passages. If you include multiple pathways that Sphero could use to make it through the maze, teams will need to determine which route will be the easiest to navigate or that they can program Sphero to move the quickest through.

- As a class build the maze that students will then program their Spheros to move through.
- Optionally, you can build the maze before the students come in (which will expedite the activity)

## PART 3: Data Gathering - 30 minutes

During this section students will have time to collect all necessary information to build their macro to navigate the maze.

- In order to know how long to program sphero to drive at different headings it will be necessary to figure out how fast it moves. To do this the students may want to make a simple Macro where Sphero moves straight forward for 1 second at 100% speed. Then the students can measure how far Sphero has moved in that second. With that information, they can determine the rate or speed at which Sphero rolls.
- Next, students should work in groups to draw the maze on a piece of paper.
- As a team, students should determine the path they plan to take through the maze and then draw it on their paper. Students may need to take measurements to help decide the best route.
- The teams then should take any necessary measurements of distances and angles in order to build a program to navigate through the maze, following the path they drew in the previous step.

## PART 4: Planning – 30 minutes

In Part 4 students convert the information they have collected on their drawing into written instructions in plain language. The development of sequential instructions will be extremely useful for programming. At the end of this phase, each group should have a drawing of way they will move through the maze and step by step instructions.

- Students should write down a list of instructions in their own words of how they will have Sphero navigate the maze.

## PART 5: Programming – 90 minutes

Time to put all of the planning to work and start programming! Students build a macro in MacroLab to move through the maze.

- In teams build a MacroLab program to navigate the maze according to the instructions written in the previous step.
- Students should be testing the macro as they go to ensure it will work when its time to send their Sphero through the maze.

## PART 6: Testing – 45 minutes

Time to perform! To make it more exciting, think about adding penalties when Sphero runs into objects in the maze or let teams who chose different paths race simultaneously.

- Each team tests their macro by moving their Sphero through the maze for time.



## PART 7: Reflection and discussion– 45 minutes

Each individual should write up their reflections on the activity. Some potential prompts and questions that you may want to ask are:

- What worked and what didn't?
- How would each student do things differently in the future?
- What route worked best?
- What was the trickiest part of the maze?
- What was the most challenging part of the activity?
- Which role did each student enjoy most and least from parts 3 and 5?

## PART 8: Share your experience on @SpheroEdu

We love seeing SPRK in action! Tweet us a few photos and we'll share them with the world!

PLAY IS A  
POWERFUL  
TEACHER