DIGITAL EDITION
Includes references to the Australian Curriculum:
Digital Technologies Content descriptions

sphero®

BOLT™

BOLT POWER PACK
EDUCATOR GUIDE

BOLT Power Pack Educator Guide by Sphero

For information about this title or to order other books and/or electronic media, contact the publisher:
Sphero, Inc
education@sphero.com
www.sphero.com

Library of Congress Control Number: 78-1-7331447-5-9

Table of Contents

Welcome

Meet Sphero BOLT

Fully programmable and highly advanced, Sphero BOLT was designed for learners of all ages. BOLT fosters a love of robotics, coding, and STEM principles—all through play-based learning. With BOLT students can:

- program with draw, block, and text languages on any type of computing device
- collect data from onboard sensors—including a compass, gyroscope, accelerometer, and light sensor
- communicate with other students via Infrared (IR) messages
- show their creativity with custom animations on the 8x8 LED light matrix

Meet your BOLT Power Pack

The BOLT Power Pack is your one-stop shop for launching and maintaining a successful STEM and programming initiative. The possibilities with your BOLT Power Pack are endless. In addition to 15 Sphero BOLT robots, the Power Pack includes:

- 15 inductive charging bases
- 15 Turbo Covers
- 15 Protractors
- 15 rolls of Maze Tape
- transportable, durable case

This Educator Guide is designed to help you get started and maximise the value of your new BOLTs in the classroom. You will:

- explore the many benefits of BOLT for student engagement and learning
- learn about the features, functionality, and use of your BOLT robots
- access eight ready-to-implement lessons that you can use with your students to launch learning
- dive into the Sphero Edu app and learn how to use it to set up and manage your BOLT classroom
- discover ways to make it all work in any classroom environment

SPHERO MISSION

Sphero is transforming PK-12 education with accessible tools that encourage exploration, imagination, and perseverance through STEAM and computer science. With the help of educators around the world, we are empowering learners of all backgrounds and abilities to discover their interests and passions while equipping them with the skills they need to be the world's future Changemakers.

Why BOLT?

Standards Aligned Learning for Grades 3 and Up

BOLT can be used to teach computer science concepts and supplement subject matter in any content area. Sphero provides extensive learning activities that are aligned to Common Core ELA and Maths, NGSS, CSTA, ISTE, and other national and international standards for easy integration into curriculums.

You, as a teacher, don't have to be a programming expert to integrate BOLT into your classroom instruction. The Sphero Edu app offers three different coding "canvases"—Draw, Block, and Text – that move from beginner to more advanced coding skills. The three coding canvases make it easy for you to progress as a programmer alongside your students.

Draw Canvas
Program BOLT with a drawing interface.

Block Canvas
Program BOLT by dragging and dropping blocks that represent lines of code.

Text Canvas
Program BOLT with the programming language JavaScript.

BOLT is truly a low-floor, high-ceiling robot. For young learners, BOLT makes getting started on their programming journeys engaging and fun. But with advanced sensors and a JavaScript Text canvas, BOLT ensures there is no limit to the possibilities for older students.

DIGITAL EDITION
Includes references to the Australian Curriculum:
Digital Technologies Content descriptions

Computational Thinking

Your students will develop important computational thinking (CT) skills when learning with BOLT through standards-aligned, purposeful lessons as well as unstructured, open play.

Computational thinking is a systematic set of processes for addressing complex problems. These processes are often described as "four pillars." The table below defines each pillar as well as provides descriptions of how learning with BOLT develops the thinking skill.

CT Pillar	DEFINITION	Example with BOLT
Decomposition	The breaking down of something—like a complex problem—into smaller, more manageable parts	Describing a programming challenge, like a maze, in parts, then programming BOLT to accomplish each discrete part in sequence
Pattern Recognition	The analysis of similar objects or ideas to extend or create patterns to better understand a problem	Analysing how three inputs—heading, speed, and duration—govern BOLT's execution of roll blocks to gain precision over the robot's movement
Abstraction	The process of weeding out the important information and ignoring irrelevant details	Using flow charts to plan, execute, and communicate a BOLT program clearly and efficiently
Algorithmic Design	The development of steps used to solve a problem, often a sequential set of rules that are followed	Programming an if…else control to execute different groups of code when a BOLT sensor gathers data within different ranges

Consider the following ways CT benefits students in computer science, other subject areas, and their everyday lives.

- **CT helps students solve problems related to computers.**

  For example, a student may decompose a difficult programming goal into smaller manageable parts using an ordinary language called pseudocode.

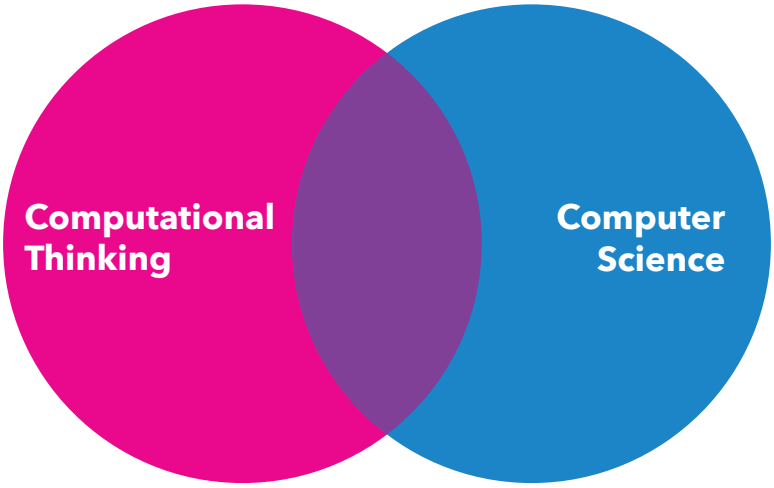
- **CT helps students apply computers to problems in the world around them.**

  For example, a student may create an algorithm to program BOLT to travel systematically around a learning space to record light levels in different parts of the room.

- **CT helps students address problems unrelated to computing.**

  For example, a student may use abstraction to hone in on important details in a book to identify the themes.

CT is commonly discussed interchangeably with computer science, but, while they are related, they have some key differences. The CT skills that your students develop while working through programming problems with BOLT will benefit them in their future programming endeavours as well as other aspects of their lives.



Computer Science for ALL Students

BOLT was designed to make learning the fundamentals of programming and computer science accessible and fun for all students, regardless of their backgrounds or abilities. As we continue to shift to a more and more digitised future in both our work and everyday lives, we will need more workers in computer-related occupations. The U.S. Bureau of Labor Statistics predicts that the number of jobs in the industry will increase by 13% from 2020-2030, faster than in other occupations.[1]

Yet, the computing field still struggles to attract and retain a workforce that accurately represents the gender, racial, ethnic, and socioeconomic diversity of our communities. A more diverse workforce—along with the rich perspectives and backgrounds that come with it—is essential to solving the computing problems of tomorrow.

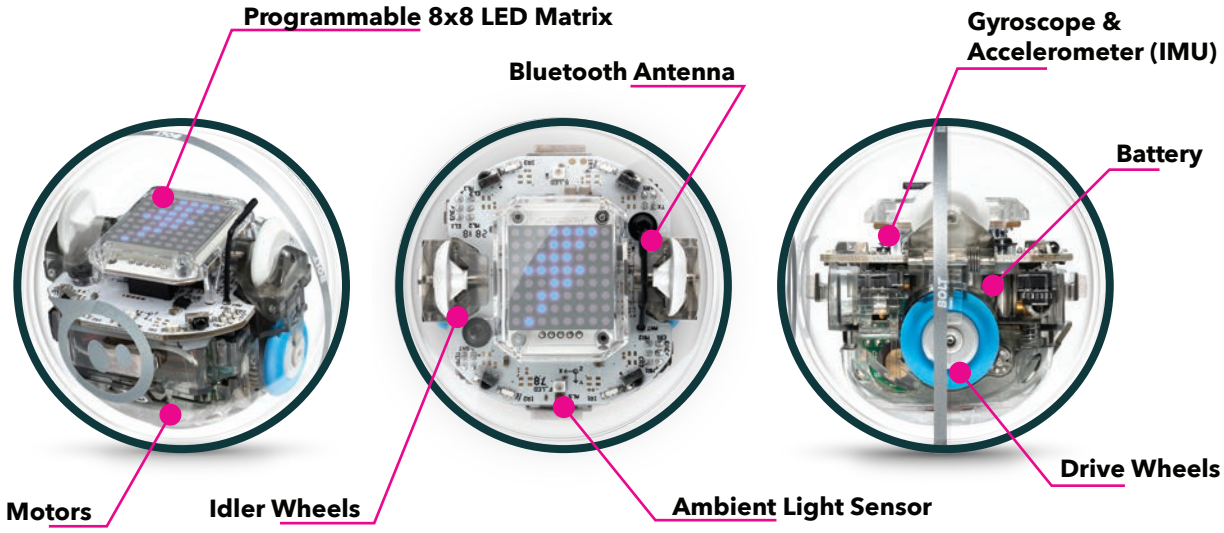
[1]US Bureau of Labor Statistics (https://www.bls.gov/ooh/computer-and-information-technology/home.htm)

The work of organisations like code.org and Girls Who Code has shown that introducing computer science to students early in their PK–12 education is key to attracting a broader array of students to computer-related professions. BOLT can make students' initial programming experiences hands-on and visual. Drag a roll block onto the canvas, execute the program, and immediately observe the effect your program has on the robot. Consider how the following active uses of BOLT make learning computer science accessible and tangible:

- using trial and error to program BOLT's path through a maze
- communicating with a learning partner to adjust the values in a comparator
- creating a game with BOLT to showcase an understanding of the pitch and roll orientation sensor
- sharing a unique story through a BOLT program with lights, sounds, and movements

Compared to more passive instructional methods, such as watching a video tutorial or listening to a teacher, learning with BOLT places the students squarely in the centre of the learning process.

Getting Started



BOLT is super durable, programmable, and extensible. Its features will help learners of all abilities in grades 3 and up develop their programming and thinking skills.

BOLT's **battery** will last an entire school day under normal usage.

BOLT is equipped with **two drive motors**, one on each side of the robot, to enable movement like rolling and spinning. Motor encoders report back data on speed and distance.

BOLT has a **programmable 8x8 LED matrix** that can display any colour, animation, scrolling text, or real-time data you tell it to! BOLT also includes a front RGB LED and a back RGB LED.

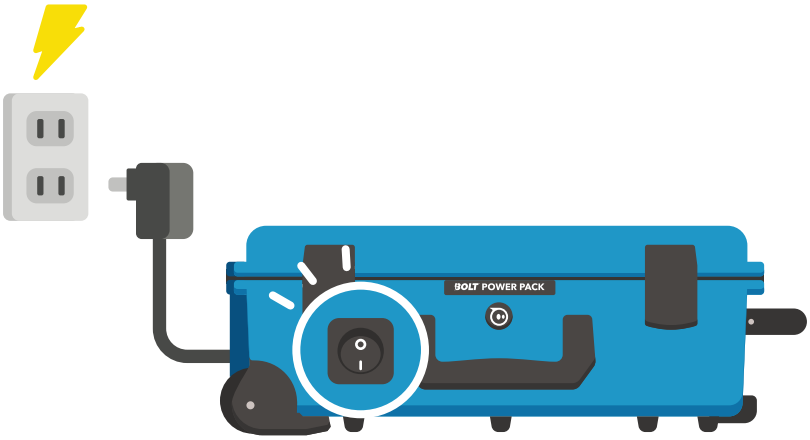
BOLT's onboard **sensors** allow it to gather data during program execution that can be used to program interesting logic in the Sphero Edu app. Here is an overview:

- **Accelerometer**: The accelerometer measures linear acceleration and can be used to detect changes in speed and collisions.
- **Gyroscope**: The gyroscope measures twisting or rotational movement around the pitch (x-axis), roll (y-axis), and yaw (z-axis).
- **Light Sensor**: The light sensor reads the light intensity (luminosity) in your environment from 0 - 100,000 lux, where 0 lux is full darkness and 30,000-100,000 lux is direct sunlight.
- **Infrared (IR) Sensors**: The IR sensors can be used to send and receive messages between BOLT robots within a 4 metre range.
- **Compass**: The compass sensor (it's really a "magnetometer") allows BOLT to know its orientation on earth, just like a normal compass.

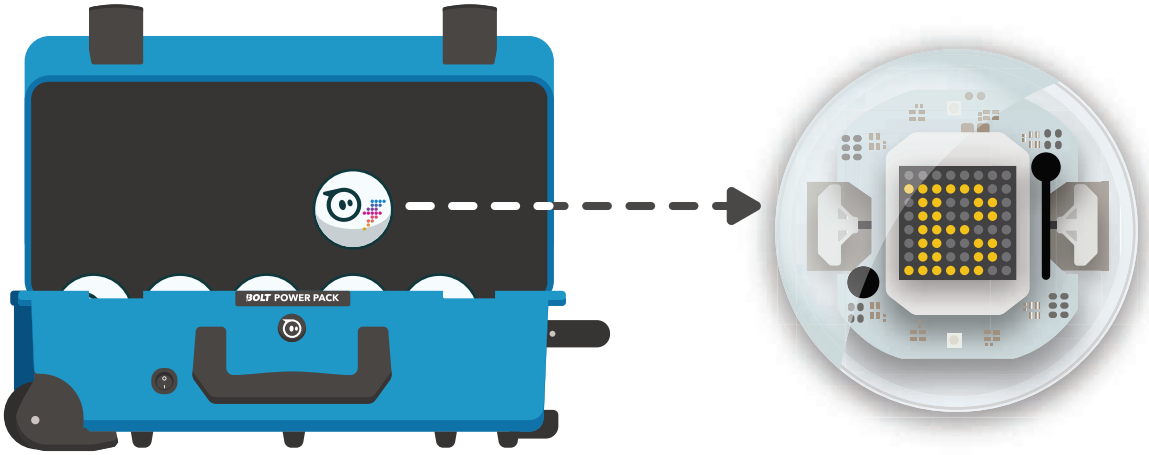
Operation

To turn on BOLT:

1. Plug your BOLT Power Pack into a wall outlet and turn the power switch on to wake up the BOLT robots.



2. Remove a BOLT from its charging cradle. The BOLT will display a lightning BOLT, then a battery charge display, then the robot name (SB-XXXX).

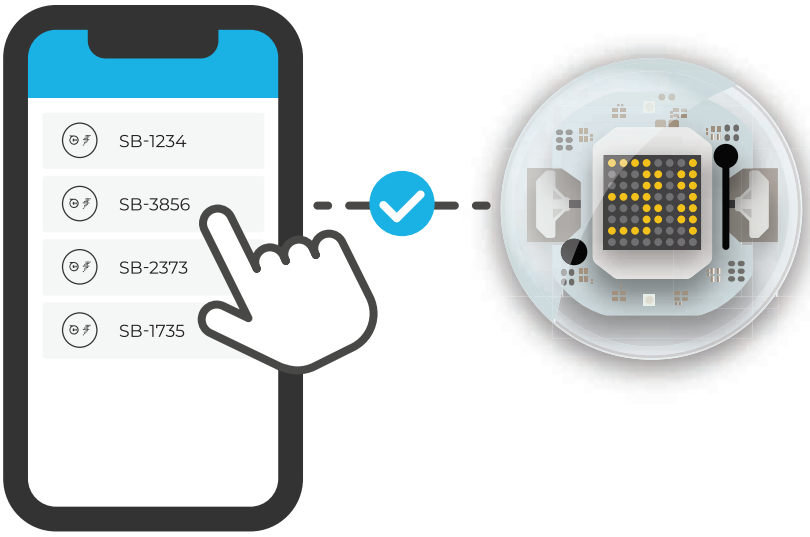


To connect to the Sphero Edu app:

1. Download the Sphero Edu app on iOS, Android, Kindle, Mac, Windows, or Chrome: sphero.cc/edu-d

2. Open the Sphero Edu app and find the 'Connect Robot' button.

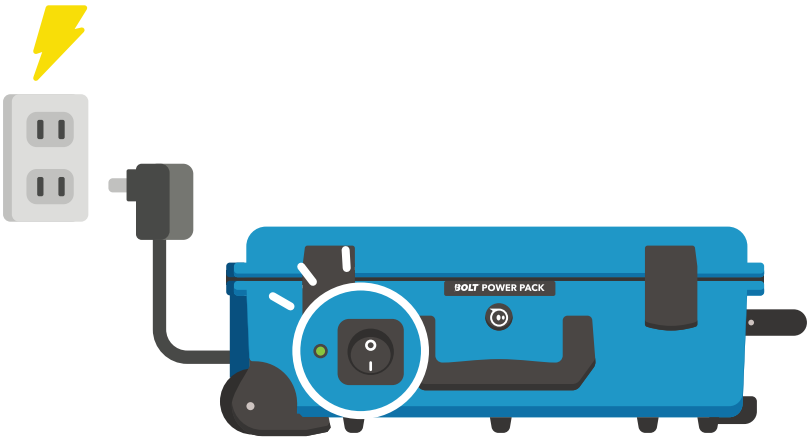
3. Hold your device near your BOLT.
4. Select BOLT from the list of robots, and then select your robot from the list to connect. Your robot name (SB-XXXX) will scroll on BOLT's 8×8 LED matrix three times after being turned on. This is the same name you will see in the Sphero Edu app robot list.



CLASSROOM TIP: To connect multiple devices to multiple BOLTs resting in the Power Pack, ask students to approach the Power Pack one at a time. Each student chooses to connect to any available BOLT on the list. When that BOLT lights up, the student removes it from the Power Pack and takes it back to their workspace. This is their BOLT robot for the day.

Charging

To charge the BOLTs in your Power Pack:



1. Place BOLT robots heavy-side down on the charging cradles inside of the Power Pack, plug the power cord in, and turn the power switch on.

Power On **Charging** **Charged**

2. The green light on the front of the Power Pack indicates the power is on. The blinking blue cradle lights indicate the robots are charging.
3. It takes six hours to charge a fully-depleted BOLT robot. Charge until the blue charger lights stop blinking. They will be solid blue when the robots are fully charged. It's OK to keep robots on the chargers for longer.

Battery Care

If you are not using the BOLT robots for a few weeks like over a winter or summer holiday, it's best to turn them off and then store them. Storing the BOLTs at approximately 50% charge in off mode will ensure the longevity of the robots' battery lives.

1. With the Power Pack on, press and hold the cradle button and simultaneously remove each BOLT to turn it off. Repeat for all robots and place them aside. You can also turn off your robot in the settings of the Sphero Edu app.
2. Toggle the power switch off and unplug the Power Pack.
3. Place robots back inside the Power Pack and store at room temperature. To turn robots on after storage, plug in and turn on the Power Pack. After storing for several months, you may need to recharge the robots.

Care

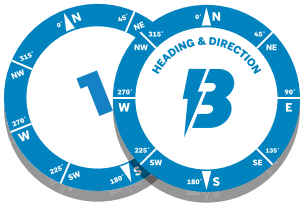
BOLT's hardware is encased in a durable polycarbonate shell and is shockproof from falls up to three feet (0.9 metres). While built to withstand falls, we don't recommend testing this theory from the top of a tall building.

BOLT is completely waterproof and doesn't have charging ports or openings to worry about. To clean and disinfect your robots:

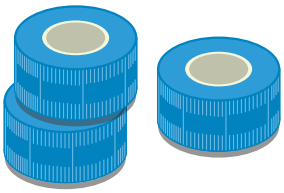
1. Use your preferred cleaning spray, gloves, paper towels, and/or disinfecting wipes (Lysol or Clorox or similar brands are best).
2. Wipe and spray away. Wipe down BOLT's outer surface. Just be sure not to use harsh solvents or anything abrasive or sharp to clean them.
3. Allow BOLT to dry completely before placing it back on in the Power Pack or on a charger.

Power Pack Accessories

In addition to 15 BOLT robots, your Power Pack comes with the following accessories to boost student learning and fun:



Protractors help students get comfortable with BOLT's heading. Place the BOLT inside the Protractor, aim the robot directly at 0° N, and then use the Protractor to plan BOLT's movement while programming.



Each 13-metre roll of **Maze Tape** includes metric unit markings (centimetres and metres). Use it to map out challenges on the floor or measure BOLT's roll movements to increase the precision in your programming.

Turbo Covers help protect BOLTs from scratches and scuffs while also providing enhanced traction. Use the Covers for better control over the robots in movement-based challenges.

The Sphero Edu App

The Sphero Edu app is designed to allow you and learners to take advantage of BOLT's features and grow with their robot as their understanding of programming develops.

The Sphero Edu app includes many features, including:

- standard-aligned learning activities for a variety of skill levels and content areas
- pre-created programs for BOLT. Execute the program as is or use it as a starter for coding your own!
- Draw, Block, and Text Canvases for creating programs
- additional information about Sphero's robots

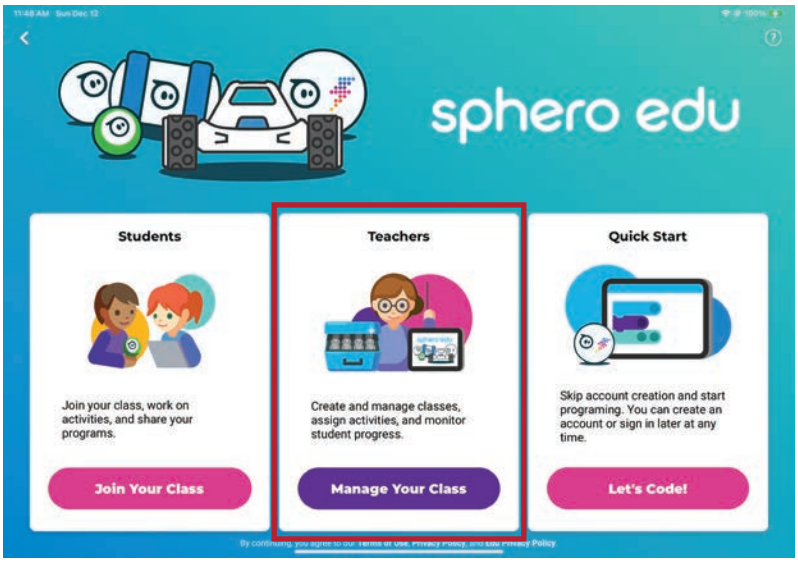
Compatible Devices

To get started, the app will need to be downloaded on student learning devices. The app is available for free in the iOS, Google Play, MacOS, Microsoft, and Amazon app stores for use on any type of classroom device. For use on Chromebooks, you will need to install the Sphero Edu Android app from the Google Play store.

Download the app for your device:

Connecting as an Educator

To get started as a teacher with the Sphero Edu app, select "Teachers" the first time you open the app.



You'll be guided through account creation. After you've created your account, you'll have access to all the same programming tools as your students. You'll also be able to log in to your account at edu.sphero.com to set up classes and assign activities to your students.

CLASSROOM TIP: It's a good idea to set up your class and invite students before they use the app. If students create accounts before a class is made, they will have to enter a guardian email to for account approval.

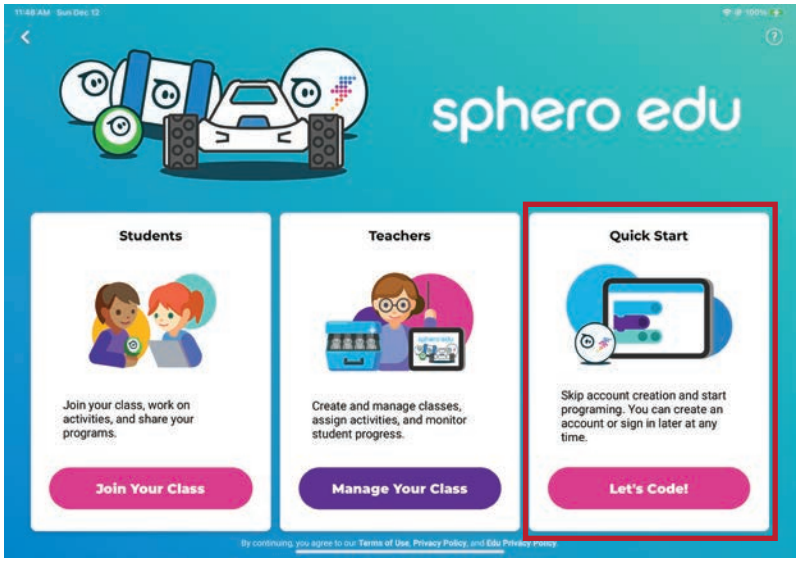
Learn more with the link below about how to set up and manage your Sphero classroom:

sphero.cc/gettingstarted

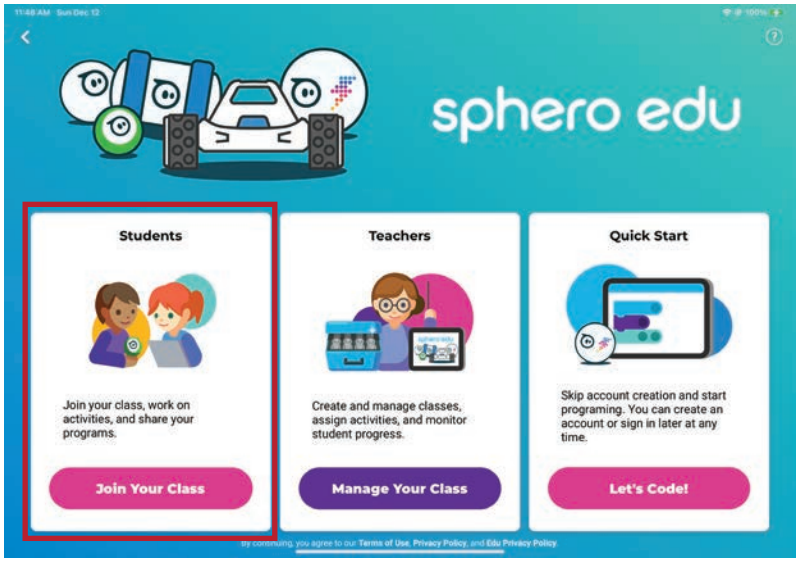
Connecting as a Student

Once the app is installed, students have two options for connecting their devices with BOLT and getting started with programming:

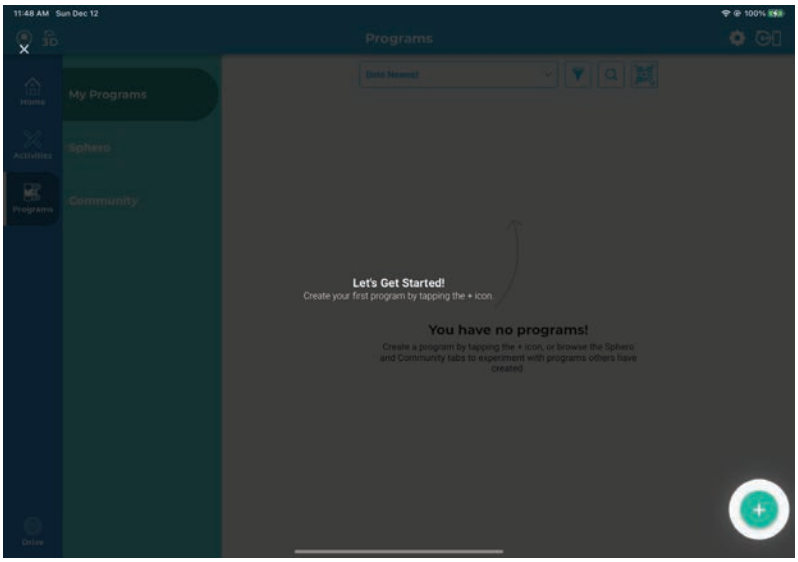
1. Select Let's "Code" in the Quick Start box to jump right in without creating an account. Students will be able to create and execute programs. However, none of their work will be saved to the cloud to be available from another device.



2. Select Join your Class in the Students box to open the app with a class code (if you've already created a class) or sign in with a Google, Clever, or Sphero account. This method will allow students to save their work to the cloud.



3. Once the app opens, choose “Programs”, then select the plus symbol in the low right corner to start your first program.



Classroom Activities

Your BOLT Power Pack is your STEAM classroom on wheels. But now that you have it, how do you get the ball rolling with STEAM learning? The eight out-of-the box, unpack-and-go activities included in this Educator Guide will help you and your students get up and running quickly with BOLT and the Draw, Block, and Text Canvases, no previous experience required. Each lesson is designed for students of all ages and abilities and can be completed in one 45-60 minute class period. Complete them in order or skip around based on your students' comfort level with BOLT and programming. While students need to have the Sphero Edu app installed on their devices, they will not need to create or sign into accounts to work on the programs.

DRAW	CONTENT FOCUS	PROGRAMMING FOCUS
Lesson 1: BOLT Boat Race	computer science	sequences
BEGINNING BLOCK		
Lesson 2: Target Maths	maths	movement blocks
Lesson 3: Help BOLT Go Home!	geography	movement, lights, and sound blocks
INTERMEDIATE BLOCK		
Lesson 4: Food Webs	science	variables and speak blocks
Lesson 5: Loopy Pictures	art	loop controls
ADVANCED BLOCK		
Lesson 6: World Games with BOLT	language arts	variables, functions
Lesson 7: BOLT Plays with Probability	computer science	if then controls
INTERMEDIATE BLOCK		
Lesson 8: Roll BOLT at the Sphero Arcade	computer science	JavaScript syntax

Before teaching:

- Read through the educator instructions for the activity, try out the programming challenge, and anticipate obstacles that your students will encounter.
- Prepare the activity area as directed in the instructions using Power Pack materials and copies of Targets from page 71. A digital version of the Targets is available to print at sphero.cc/targets.
- Copy and/or prepare to project the student-facing instructions for the lesson.
- Make sure all robots and programming devices are fully charged.
- Plan for two students per BOLT robot.

During teaching:

- Use the Exploration and Skills Building steps to launch the activity and introduce foundational skills.
- Give students the opportunity to engage with the Challenge without guidance. Circulate the learning space to celebrate teamwork, mistakes, and problem solving.
- Encourage students to extend their learning with the Extended Challenge options.

After teaching:

- Reflect on student learning. What was easy for students? What was more difficult? Was there an artifact of student learning that you can share with the whole class to address a common challenge or showcase a programming concept or problem solving strategy?
- Plan your next lesson. Use another lesson in this Educator Guide or try another Sphero activity in the Sphero Edu app.

Lesson 1: BOLT Boat Race

Overview
BOLT doesn't need a sail to compete in a boat race! In this lesson, students learn to program and control BOLT's movement with the Draw Canvas.

Programming Level
Beginner Draw

Learning Objectives
1. I can use the Draw Canvas to program BOLT to roll in different directions.
2. I can use the Draw Canvas to program BOLT to roll at different speeds.
3. I can use the Draw Canvas to program BOLT to change colours.

Vocabulary
- **Draw Canvas:** The editor in the Sphero Edu app for creating programs by drawing lines
- **Execute:** To run or complete a block or program

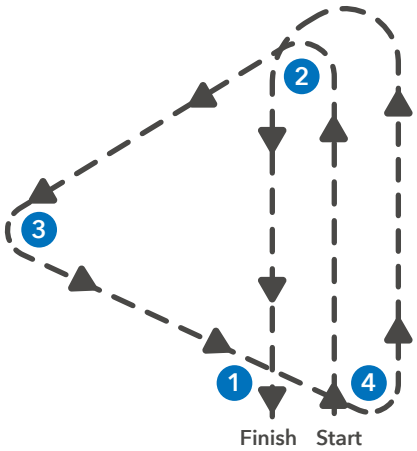
Reference to the Australian Curriculum: Digital Technologies Content descriptions
- Year 3-4 Implement simple digital solutions as visual programs with algorithms ~~*involving branching (decisions)*~~ and user input (ACTDIP011)
- NOTE: This lesson does not address all aspects of this particular Content description, and these are indicated by the italicised wording above.

Content Connections
- computer science

Materials
- BOLT Power Pack
- Programming devices with the Sphero Edu app installed
- Targets (page 71)

Preparation
- Set up four to five racecourses around your learning space. Each racecourse will need a 6×6 ft square area. You could use the centre punchout from the Protractor to mark buoys or label Targets printed or copied from page 71 of this guide.
- Preview and prepare to share the Student Instructions (page 28).

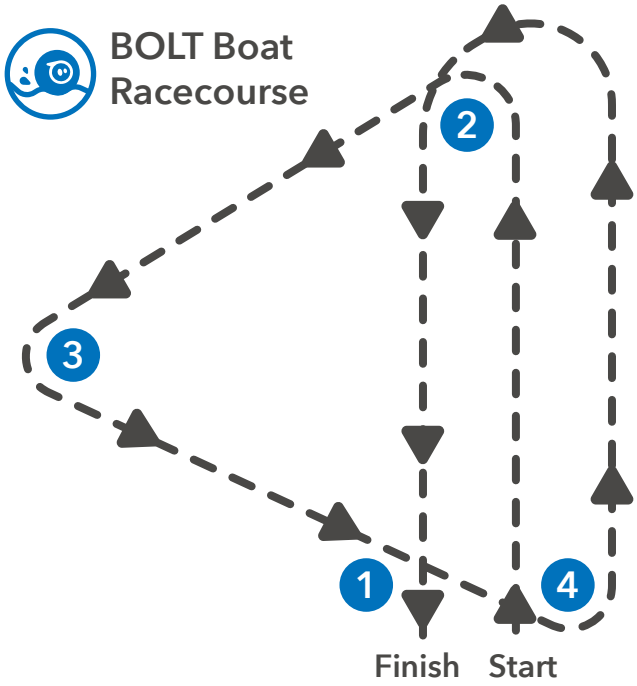
Setup



Program QR Code
sphero.cc/BOLT1

Exploration

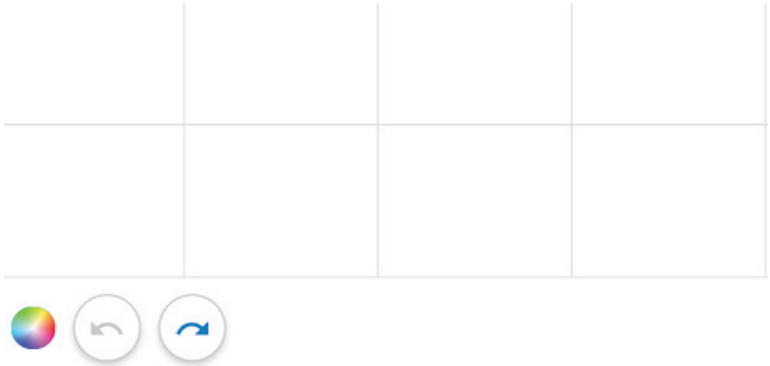
1. Introduce your students to sailing as an Olympic sport. Boats race around buoys in a specific order. The first one to cross the finish line wins! If possible, show a video to introduce students to sailing races like this one from the 2020 Tokyo Olympic Games (sphero.cc/fc008b).



2. Use the QR code to open a blank Draw Canvas. Review the following with students:
   - Connect a BOLT robot with their programming device.
   - Aim their BOLT so the bluetaillight is pointing directly at them.
   - Drive their robot.
3. Ask students to practice driving their BOLT through the racecourse. BOLTs should start in between Targets 1 and 4, drive around the outside of Target 2, then Target 3, then Target 4, and finally around Target 2 before finishing between Targets 1 and 4.
4. *Discuss: What makes completing a racecourse challenging for sailboats?*
   - Review, that in a sailing race, the biggest challenge is changing your boat's direction based on the direction of the wind. In a BOLT sailing race, the biggest challenge is to control the BOLT with programming instructions on the Draw Canvas.

Skills Building

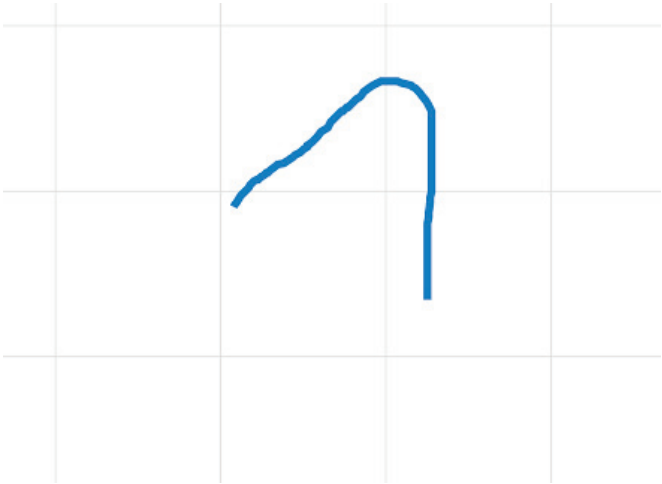
1. Ask students to open the Draw Canvas. Show students how to set the line colour and speed by selecting the colour wheel in the bottom left-hand corner of the screen.
   - Have them choose a colour for their BOLT boat and set the speed to a low setting.



2. Challenge students to use the Draw Canvas to program their BOLT boat to sail to Target 2.
   - Draw a line.
   - Select Start to execute (run) their program.
   - *Were they successful?* Ask students to iterate and refine their program until they are successful.

TEACHER TIP: BOLT moves relative to its starting position in the path of the lines on the Draw Canvas. Where you start drawing a line does not affect BOLT's physical starting location.

3. Next, challenge students to drive around Target 2 and towards Target 3 using a line with a curve. Starting slow with a small number of lines and Targets will help your students develop fluency with the Draw Canvas.



Challenge

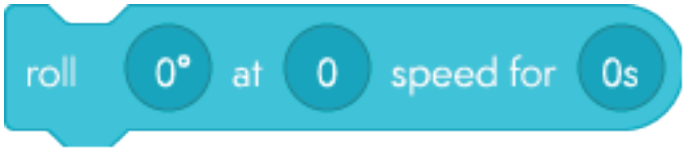
1. Set up your class BOLT boat races.
   - Group students with 3-4 BOLTs to a sailing race course.
   - Ask each group to choose a unique colour so they can tell their BOLTs apart.
   - Tell students to program their BOLT and then start their programs at the same time.

TEACHER TIP: If students have a hard time navigating the course in one setting, ask students to execute one line at a time. By starting and stopping the program, they should eventually be able to make it through the course.

2. *Which BOLT wins?* As time allows, switch student groups around the room so students can race as many of their classmates as possible. Consider setting up a competition bracket to see who is the ultimate class champion.
3. *Discuss: What are three things you learned about the Draw Canvas?*

Extended Challenge

1. If your students are feeling comfortable with the Draw Canvas, it may be time to explore the Block Canvas. Invite students who are ready to navigate their BOLT boats through the racecourse using **roll blocks** and **delay blocks**.
   - **roll blocks** define the heading, speed, and duration of BOLT's movement.



   - **delay blocks** allow the BOLT to pause in between each roll.

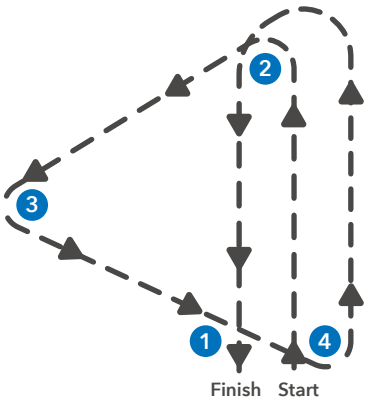
Lesson 1 : BOLT Boat Races

BOLT doesn't need a sail to compete in a boat race! In this lesson, you'll learn to program and control BOLT's movement with the Draw Canvas.

Learning Objectives

1. I can use the Draw Canvas to program my Sphero robot to roll in different directions.
2. I can use the Draw Canvas to program my Sphero robot to roll at different speeds.
3. I can use the Draw Canvas to program my Sphero robot to change colours.

Setup



Program QR Code

sphero.cc/BOLT1

Exploration:

Open the Draw Canvas or scan the above QR Code, pair your BOLT to your programming device, and practice driving the racecourse.

Skills Building:

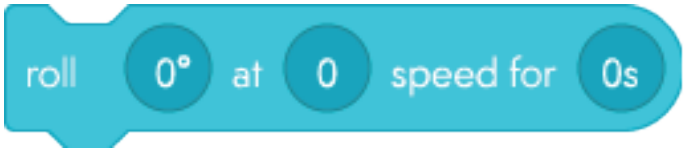
Set your programming colour, then practice control BOLT's movement with lines on the Draw Canvas.

Challenge:

Race your classmates through the racecourse and test your Draw Canvas programming skills.

Extended Challenge:

Ready for more? Try programming your BOLT Boat on the Block Canvas with **roll** and **delay blocks**.



Lesson 2: Target Maths

Overview

Play a maths operations game and learn how to control BOLT's movement in the process. In this lesson, students will use **roll** and **delay blocks** to program BOLT to roll to two numbers and use maths operations to hit a target number.

Programming Level

Beginning Block

Learning Objectives

1. I can program **roll blocks** to move BOLT to desired locations.
2. I can use the matrix character and speak blocks to animate maths equations.

Vocabulary

- **Block Canvas**: The editor in the Sphero Edu app for creating block programs
- **Input**: Place to add numeric or text values into a programming block or text statement
- **Speed**: Input to program how quickly (or slowly) a Sphero robot will move
- **Heading**: Input to program the direction a Sphero robot is pointed during or before a roll, between 0° and 360°
- **Duration**: Input to program the amount of time that BOLT will execute a movement like a roll or a spin in seconds

Reference to the Australian Curriculum: Digital Technologies Content descriptions

- Year 3-4 Implement simple digital solutions as visual programs with algorithms ~~*involving branching (decisions)*~~ and user input (ACTDIP011)
- NOTE: This lesson does not address all aspects of this particular Content description, and these are indicated by the italicised wording above.

Content Connections

- computer science
- maths

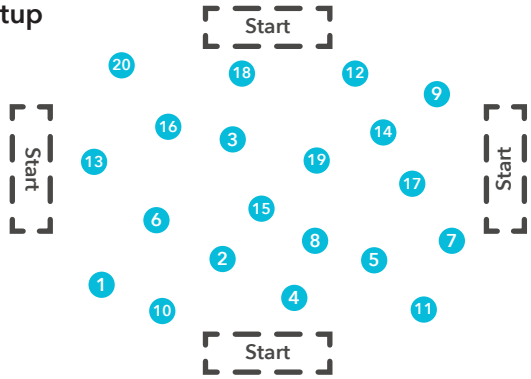
Materials

- BOLT Power Pack with robots fully charged
- Programming devices with the Sphero Edu app installed
- Targets (page 71)
- Protractors (included with BOLT Power Pack)

Preparation

- Clear a 20' by 20' space to set up the activity. If your classroom space is limited, consider conducting this lesson in a gym, cafeteria, or other available multi-purpose room. Print out or copy approximately 20 targets from page 71, label them 1-20, and spread them around the available area. Designate four start locations around the numbers.
- Preview and prepare to share the student instructions (page 33).

Setup



Program QR Code

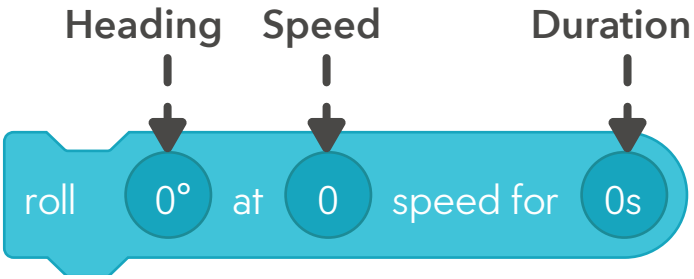
sphero.cc/BOLT2

Exploration

1. Warm up students by playing a few rounds of Target Maths without BOLT.
   - Announce a target number less than 20.
   - Ask students to think of a maths expression that "hits"—or equals—the target number.
   - For example, if you choose 15, students could suggest 5 + 10, 16 - 1, or 5 × 3.
2. Show students the Target Maths playing area and ask students to find the numbers they used in their maths expressions.

Skills Building

1. Explain that before students can program their robot to roll to two numbers, they have to be able to use a **roll block** to program BOLT to roll to one number.
2. Discuss the three inputs in the **roll block**.
   - **Heading** is the direction BOLT is pointed between 0° and 360°. 0° is the aim direction, facing directly away from the blue taillight.
   - **Speed** is a number between 0 and 255 which controls how fast BOLT rolls.
   - **Duration** is the time in seconds for a roll.

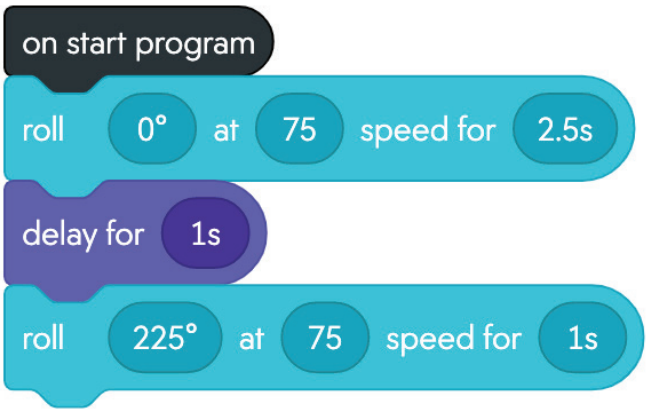


3. Give students an opportunity to practice controlling BOLT's movement with **roll blocks**. Ask students to do the following:
   - Place their BOLT in a start location.
   - Choose a target number.
   - Aim BOLT directly at their number.
   - Connect a **roll block** to **on start program**.
   - Adjust the speed and duration to make BOLT stop near their target. Anywhere within 12" of the target is a great roll!

TEACHER TIP: There will likely be some BOLT collisions as students try to program their BOLTs to roll to different numbers. Make these "bumps" part of the game. If a collision takes place, students can bring their BOLTs back to start and try again.

Challenge

1. Play Target Maths with BOLT.
   - Announce a target number.
   - Ask students to program their BOLT to roll to two numbers that can be used in a maths expression to hit the target.
   - If students want to use the same number twice, instruct them to roll away from and then back to the number.
2. Show students how to use a **delay block** in between every **roll bock** to make sure that BOLT comes to a complete stop.



TEACHER TIP: The heading in the second roll block can be tricky for students to figure out. Place a Protractor over a BOLT robot with 0° pointed in the aim direction. Now students can see the directions of headings easily from 0° to 360° in relation to BOLT's orientation.

3. As time allows, play a few rounds with different numbers.

Extended Challenge

1. Once your students are confident using **roll** and **delay blocks**, they'll be ready for more. Prompt them to add some of the following challenges:
   - Use a **matrix character block** to show operations while rolling to the second number.
   - Use **speak** and **scroll text blocks** to announce the full equation after they've rolled to the second block.
   - Make expressions with three or more numbers to equal a target number.

INTERESTED IN MORE Maths ACTIVITIES?

Try out some of the Sphero maths activities listed on the STEAM Activities and Skill Progression Chart (sphero.cc/chart).

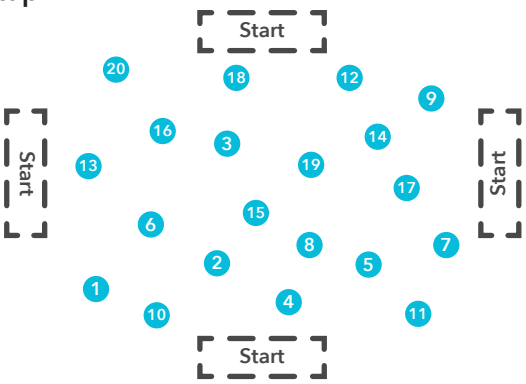
Lesson 2: Target Maths

Play a maths operations game and learn how to control BOLT's movement in the process. In this lesson, you'll use **roll** and **delay blocks** to program BOLT to roll to two numbers and use maths operations to hit a target number.

Learning Objectives

1. I can program roll blocks to move BOLT to desired locations.
2. I can use the matrix character and speak blocks to animate maths equations.

Setup



Program QR Code

sphero.cc/BOLT2

Exploration:

Play a round or two of Target Maths without BOLT. Make maths expressions out of two other numbers to "hit"—or equal—a target number. For example, if the target number is 15, you could use 5 + 10, 16 - 1, or 5 × 3.

Skills Building:

Practice using **roll blocks** to control BOLT. Place your BOLT in a starting location, choose a target number, aim your BOLT at your number, and then adjust the speed and duration in a **roll block** to program BOLT to roll to your number.

Challenge:

Now play Target Maths with BOLT. Use **roll** and **delay blocks** to program BOLT to roll to two numbers that can hit the target number.

Extended Challenge:

Ready for more? Try some of the following:

- Use a **matrix character block** to show operations while rolling to the second number.
- Use **speak** and **scroll text blocks** to announce the full equation after you've rolled to the second block.
- Make expressions with three or more numbers to equal a target number.

Lesson 3: Triangulation

Overview

BOLT is lost in the wilderness. Luckily, it recognises two familiar landmarks in the distance: Sphero Mountain and the Programming Plateau. In this lesson, students will use the Protractor to triangulate their position, find their way home, and learn about asynchronous and synchronous programming in the process.

Programming Level

Beginning Block

Learning Objectives

1. I can use the Protractor to navigate with BOLT.
2. I can use asynchronous and synchronous programming to control how a program executes.

Vocabulary

- **Asynchronous:** Programming that executes multiple blocks or commands at the same time
- **Synchronous:** Programming that executes one block or command at a time

Reference to the Australian Curriculum: Digital Technologies Content descriptions

- Year 3-4 Implement simple digital solutions as visual programs with algorithms ~~*involving branching (decisions)*~~ and user input (ACTDIP011)
- NOTE: This lesson does not address all aspects of this particular Content description, and these are indicated by the italicised wording above.

Content Connections

- geography
- maths
- computer science

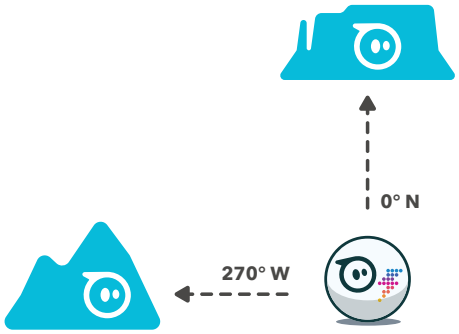
Materials

- BOLT Power Pack with robots fully charged
- Programming devices with Sphero Edu app installed
- Protractors
- Targets

Preparation

- Plan for student work areas around the learning space. Each student group will need an approximately 5′ by 5′ area of floor space.
- Print out or copy targets to mark locations. You can also draw sketches of landmarks on pieces of scrap paper.
- Preview and prepare to share the student instructions (page 39).

Setup



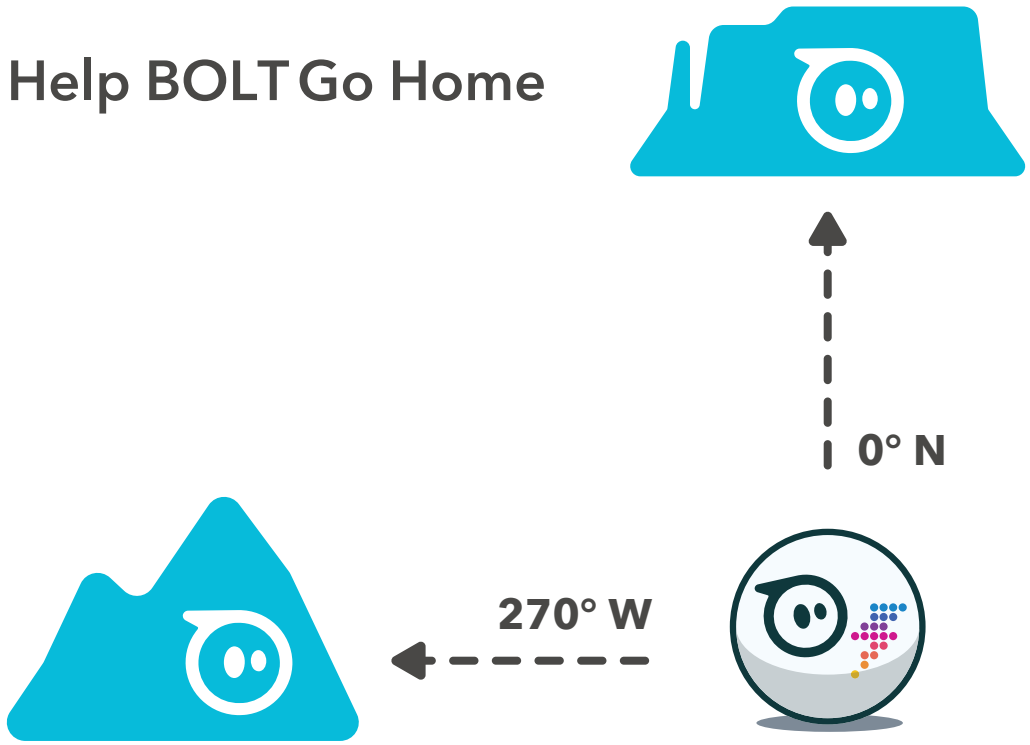
Program QR Code

sphero.cc/BOLT3

Exploration

1. Ask students which tools they'd like to have in their backpacks if they were lost in the woods. Students will likely suggest water, food, a jacket, and other survival tools. Lead them towards suggesting a compass, a valuable navigation tool especially if a phone or GPS runs out of battery or service.
2. Set the scene for this programming challenge:

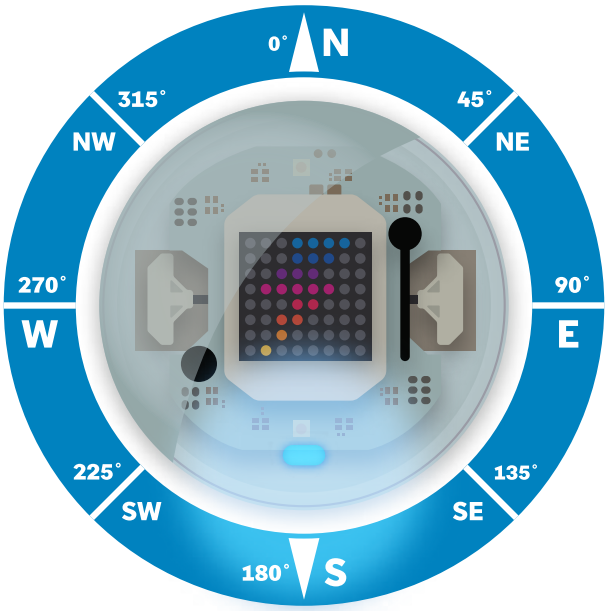
BOLT is lost deep in the woods and needs help finding its way home. Luckily it recognises two distinct landmarks. Sphero Mountain is 270° directly to the West. The Programming Plateau is directly North at 0°.



BOLT knows that its home is 315° NW of Sphero Mountain and 270° directly West of Programming Plateau. Can you help it find its way home?

Skills Building

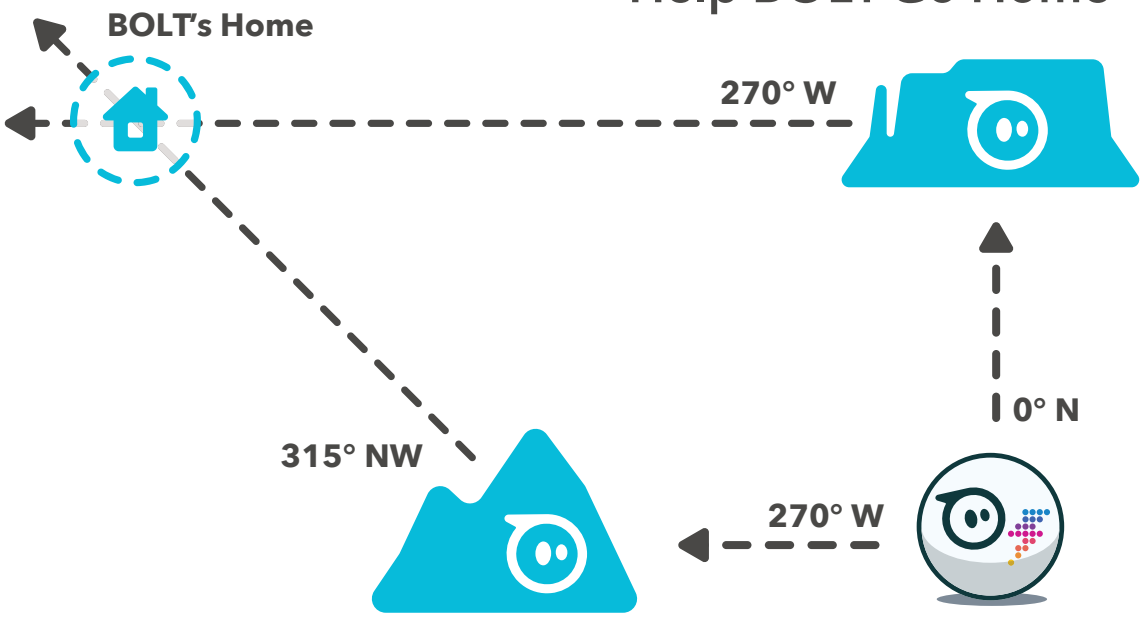
1. Together with students, set up their map.
   - Put BOLT on the ground and pair it to a programming device.
   - Aim BOLT directly away from the programmer; the blue taillight should point directly at the programmer.
   - Put the Protractor over BOLT. Align 0° North with Aim direction so that the blue light aligns with 180 degrees South.



   - Use Targets or scrap pieces of paper to mark Sphero Mountain directly to the west at 270° and Programming Mountain to the North at 0°. The distance from the BOLT is not important.
2. Challenge students to place BOLT's home 315° NW of Sphero Mountain and 270° directly West of Programming Plateau on the map. Mark BOLT's home with a Target or a piece of paper.

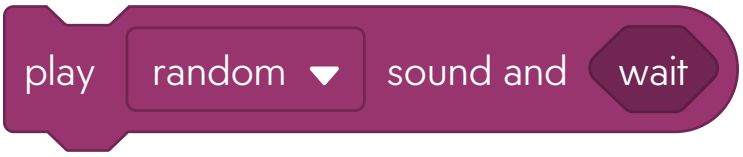
TEACHER TIP: Support students by hinting that they need to take the Protractor off BOLT and place it on or near the mountain and then on or near the plateau to find the lines that BOLT's home is on. BOLT's home is where these lines cross.

Help BOLT Go Home



Challenge

1. Now that BOLT has the lay of the land, invite students to program its way home.
   - Drag the **play adventure beginning sound block** to o**n start program**.
   - Place the **roll block** underneath.
   - Set the heading, speed, and duration.
   - Select **Start** to execute the program
2. Challenge students to get BOLT to roll while playing the adventure music. They will need to toggle **wait** to **continue** in the **play sound block**.



3. Use this opportunity to introduce asynchronous versus synchronous programming.
   - Asynchronous programming is when multiple commands execute at the same time. Toggling **wait** to **continue** tells BOLT to execute the **play sound block** and the **roll block** at the same time.
   - Synchronous programming is when commands are executed one at a time. When the **play block** was set to **wait**, the **roll block** would not execute until the adventure sound finished.

Extended Challenge

4. Once your students have navigated asynchronous and synchronous programming, they'll be ready for more. Suggest some of the following challenges:
   - Program BOLT to announce, "I'm home!" after it arrives home with a **speak block**.

     Hint: Students will need to use a **delay block** to wait until the adventure sound has finished. BOLT can only play one sound at a time.
   - Program a celebratory animation using the **matrix animation block**.

INTERESTED IN MORE SOCIAL STUDIES ACTIVITIES?

Try out some of the Sphero social studies activities listed on the STEAM Activities and Skill Progression Chart (sphero.cc/chart).

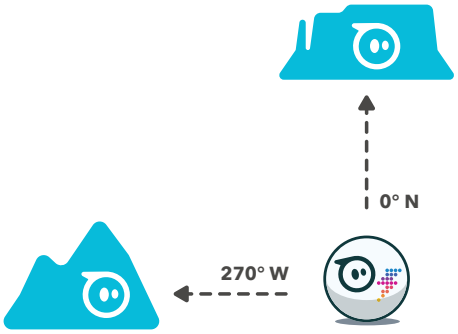
Lesson 3: Triangulation

BOLT is lost in the wilderness. Luckily, it recognises two familiar landmarks in the distance: Sphero Mountain and the Programming Plateau. In this lesson, you'll use the Protractor to triangulate BOLT's position, find its way home, and learn about asynchronous and synchronous programming in the process.

Learning Objectives

1. I can use the Protractor to navigate with BOLT.
2. I can use asynchronous and synchronous programming to control how a program executes.

Setup



Program QR Code

sphero.cc/BOLT3

Exploration:

Oh, no! BOLT is lost deep in the woods and needs help finding its way home. Luckily it recognises two distinct landmarks. Sphero Mountain is 270° directly to the West. The Programming Plateau is directly North at 0°.

Skills Building:

Make a map on the floor showing BOLT's location. BOLT also knows its home is 315° NW of Sphero Mountain and 270° directly West of Programming Plateau. Add BOLT's home to the map.

Challenge:

Use the blocks on the Block Canvas to program BOLT to roll home while playing some adventure music.

Extended Challenge:

Ready for more? Try some of the following:

- Program BOLT to announce, "I'm home!" after it arrives home with a **speak block**.
- Program a celebratory animation using the **matrix animation block**.

Lesson 4: Food Webs

Overview

Everything in nature is connected! Few models show this better than a food web. In this lesson, your class will construct a food web on the classroom floor and then program BOLT to show the food chains. In the process, your students will learn about how to use a variable to keep track of the different trophic levels in your web.

Programming Level

Intermediate Block

Learning Objectives

1. I can use BOLT to model a food web of an ecosystem that I know well.
2. I can use a **number variable** to keep track of the levels in a food chain.

Vocabulary

- **Variable**: A piece of information, like a number value, that can change during a program
- **String**: Phrases that combine letters, numbers, and/or other characters

Reference to the Australian Curriculum: Digital Technologies Content descriptions

- Year 5-6 Implement digital solutions as simple visual programs involving ~~*branching, iteration (repetition), and*~~ user input (ACTDIP020)
- NOTE: This lesson does not address all aspects of this particular Content description, and these are indicated by the italicised wording above.

Content Connections

- computer science
- science

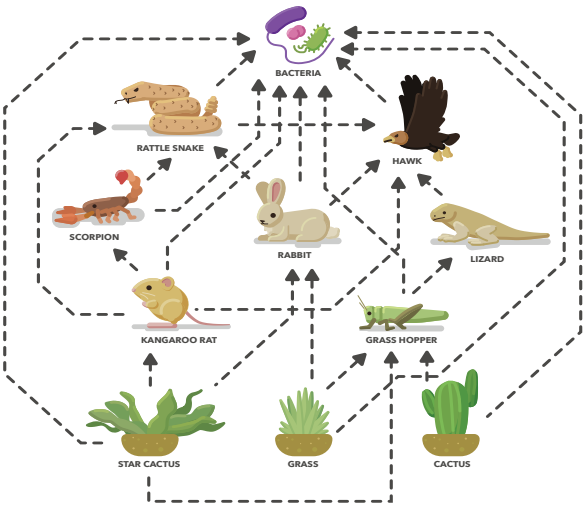
Materials

- BOLT Power Pack with robots fully charged
- Programming devices with Sphero Edu app installed
- Targets (page 71)

Preparation

- Clear a 15′ by 15′ area of your learning space to set up the food web and BOLT activity.
- Copy or print approximately 16 Targets from page 71.
- For background information on food webs and trophic levels, refer to this National Geographic Encyclopedic Entry (sphero.cc/bb9260)
- Preview and prepare to share the student instructions (page 45).

Setup

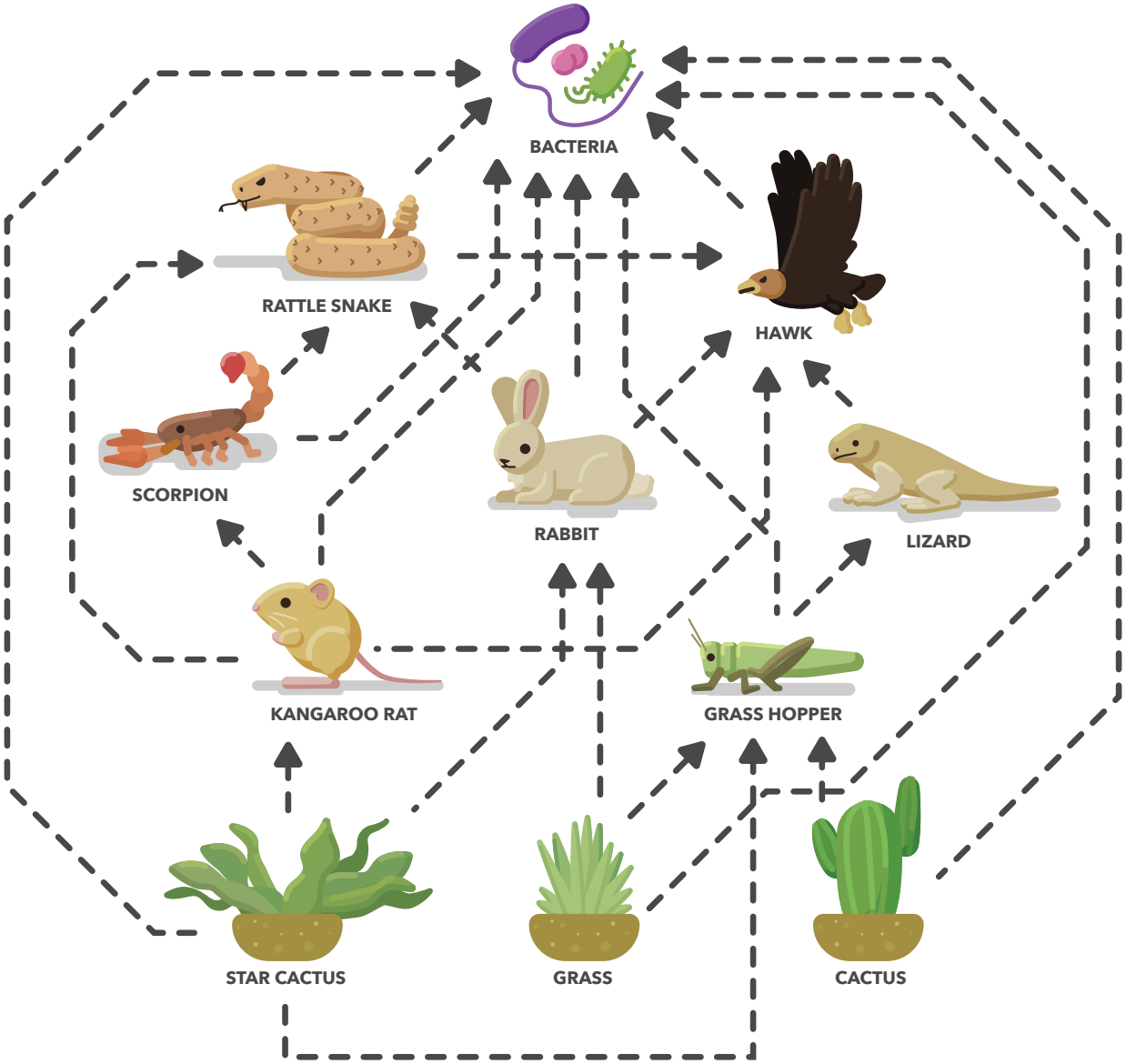


Program QR Code

sphero.cc/BOLT4

Exploration

1. Together with students, construct a food web.
   - Choose an ecosystem.
   - Brainstorm organisms, including producers and consumers, in that ecosystem.
   - Record each organism on a Target.
   - Place the Targets in a large circle.



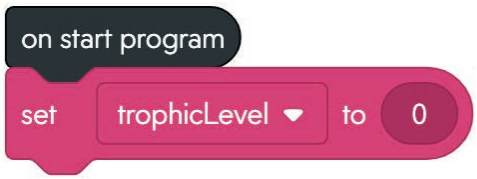
2. As a class, discuss some food chains in the food web. For example: Cactus → Grasshopper → Lizard → Hawk.

TEACHER TIP: As you create the food web, introduce or review the science-specific vocabulary that relates to this lesson. A **food chain** is a connection that shows which organisms eat other organisms in an ecosystem. Together, all of the overlapping food chains create the **food web** for an ecosystem. **Trophic levels** describe an animal's role in a food web. Producers are plants that form the bottom level. **Primary consumers** are herbivores that eat the producers. **Secondary consumers** are omnivores or carnivores that eat the primary consumer. **Tertiary consumers** are omnivores and carnivores that eat the secondary consumers.

3. Explain to students that in this lesson, BOLT will roll through the food web and keep track of trophic levels.

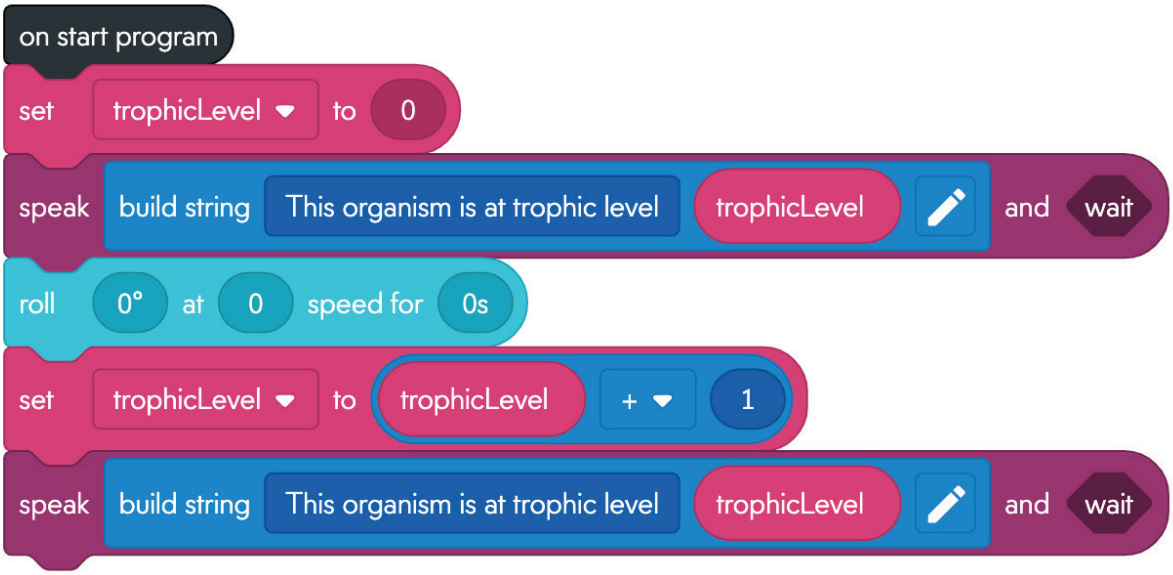
Skills Building

1. Ask students to open the Block Canvas associated with this activity.
2. Introduce the **number variable**, **trophicLevel**.
   - Explain that a variable is a piece of information, in this case a number, that can change.
   - This variable will keep track of the trophic levels in the food chain. At the start of the lesson, it is set to 0 to represent producers.



3. Discuss the **speak block**. The **speak block** will say, "This organism is at trophic level 0." At this point in the program, trophicLevel is equal to 0.

4. Invite students to program BOLT to move from a producer to a primary consumer in the food web.
   - Place BOLT on a producer in the food web (level 0).
   - Adjust the values in the roll Block to make their BOLT move to an animal that eats their producer.
5. Discuss with students: *What does BOLT say when executing the second speak block? Why does BOLT say this?*
   - The **set trophicLevel to trophicLevel + 1** immediately after the **roll block** increases the value of the variable by 1. Programmers call this "incrementing the variable".



Challenge

1. Challenge students to program an entire food chain all the way from a producer to the highest level consumer. Students can:
   - Add **roll blocks** to program BOLT's path to a new organism.
   - Continue **setting trophicLevel to trophicLevel + 1** and announcing the trophic level when BOLT arrives at each organism.
2. Discuss: *What food chain has the fewest trophic levels? The most trophic levels?*

Extended Challenge

1. If your students have additional time, prompt them to try some of the following:
   - Add **sound** and matrix animation blocks to represent different organisms in a food chain.
   - Use **speak blocks** to reveal facts about organisms one at a time.
   - Add a decomposer organism that takes the energy back to the producers so your program can loop.

INTERESTED IN MORE SCIENCE ACTIVITIES?

Try out some of the Sphero science activities listed on the STEAM Activities and Skill Progression Chart (sphero.cc/chart).

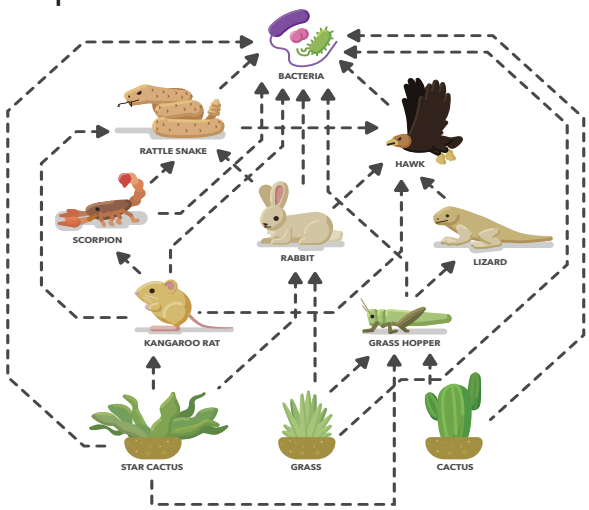
Lesson 4: Food Webs

Everything in nature is connected! Few models show this better than a food web. In this lesson, you will construct a food web and then program BOLT to show the food chains. In the process, you'll learn about how to use a variable to keep track of the different trophic levels in your web.

Learning Objectives

1. I can use BOLT to model a food web of an ecosystem that I know well.
2. I can use a **number variable** to keep track of the levels in a food chain.

Setup



Program QR Code

sphero.cc/BOLT4

Exploration:

Construct a food web with your classmates. Choose an ecosystem. Then, brainstorm organisms, including producers and consumers, in that ecosystem. Record the name of each organism on a BOLT Target. Place the Targets in a large circle.

Skills Building:

Analyse the program and figure out how the program uses a **number variable**, **trophicLevel**, and a **speak block** to announce the trophic level of organisms in your food chains.

Challenge:

Program BOLT's path through a food chain. *What food chain can you program with the fewest trophic levels? The most trophic levels?*

Extended Challenge:

Ready for more? Try some of the following:

- Add **sound** and **matrix animation blocks** to represent different organisms in a food chain.
- Use **speak blocks** to reveal facts about organisms one at a time.
- Add a decomposer organism that takes the energy back to the producers so your program can loop.

Lesson 5: Loopy Pictures

Overview

Connecting dots to make pictures is usually done with a pencil and paper. In this lesson, your students will make their pictures come to life by programming BOLT to connect the dots. Then they'll explore **loop controls** to make their programs repeat their code and go on and on and on…

Programming Level

Intermediate Block

Learning Objectives

1. I can use **loop forever** and **loop x times blocks** to execute sections of my program over and over again.
2. I can program BOLT to trace the outline of a picture I design.

Vocabulary

- **Control:** A programming tool used to make certain parts of a program repeat or execute under certain conditions
- **Loop:** A programming control that makes an action of repeating over and over again

Reference to the Australian Curriculum: Digital Technologies Content descriptions

- Year 5-6 Implement digital solutions as simple visual programs involving ~~*branching*~~, iteration (repetition), and user input (ACTDIP020)
- NOTE: This lesson does not address all aspects of this particular Content description, and these are indicated by the italicised wording above.

Content Connections

- computer science
- art

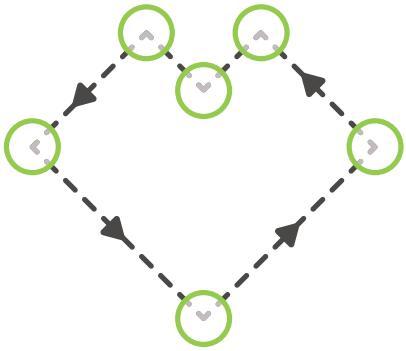
Materials

- BOLT Power Pack with robots fully charged
- Programming devices with the Sphero Edu app installed
- Targets (page 71)

Preparation

- Plan for student work areas around the learning space. Each student group will need at least a 5′ by 5′ area of floor space.
- Print out or copy Targets (8 per group). Then set up some of the Targets to make the corners of one loopy picture ahead of time.
- Preview and prepare to share the student instructions (page 50)

Setup



Program QR Code

sphero.cc/BOLT5

Exploration

1. Explain that, in this lesson, students will use Targets to create shapes and then program BOLT to trace the outline of the shape. The goal is for the outlines to be obvious enough that another group can guess their shape.
2. Divide students so that each group has eight Targets. You can combine groups so that each outline has multiple BOLTs.
3. Give students a few minutes to use Targets to create their shapes.
   - Students can number the Targets so the order is clear.
   - Keep Targets within a 5′ by 5′ work area.

TEACHER TIP: If your student designs are too complex, they will struggle to accurately program the path. To help students create simple designs, handout sets of tangrams (if available) and ask students to use eight or fewer Targets.

Recommended

Not Recommended

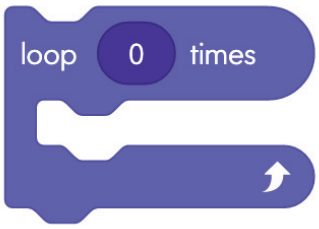
Skills Building

1. Ask students to program BOLT to connect the dots and trace the outline.
   - Use the **main LED colour block** to choose a colour for BOLT.
   - Use **roll blocks** to make BOLT roll from Target to Target.
   - Use one second **delay blocks** to make sure BOLT comes to a full stop at each Target.
2. Show students the **loop forever block** located in the control category of the block library.

- Invite students to predict its programming purpose.
- Ask students to place a **loop forever block** under **on start program** and then place all of the other blocks inside and explore what happens.
- Discuss with students what they learned about a **loop forever block**.

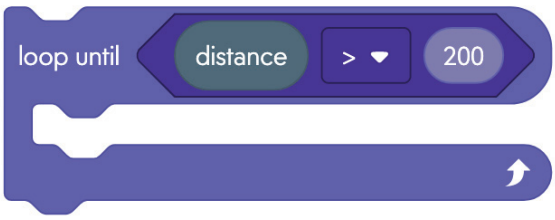
Challenge

1. Introduce the challenge: Change up your program so that BOLT connects the dots and traces your outline exactly three times.
   - Students need to replace the **loop forever block** with a **loop x times block** and set the input to 3. Allow students to explore the Block Canvas to find the solution on their own.

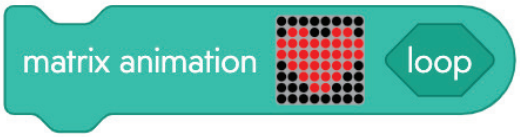


Extended Challenge

2. If your students have additional time, prompt them to try some of the following:
   - Explore the **loop until block**. Challenge students to change the program so that BOLT stops after time elapsed is greater than 20 or distance is greater than 200.



- Use the **matrix animation block** to change your BOLT shape into a light show. Students can choose one like the beating heart or play around with programming their own.



INTERESTED IN MORE ART ACTIVITIES?

Try out some of the Sphero art activities listed on the STEAM Activities and Skill Progression Chart (sphero.cc/chart).

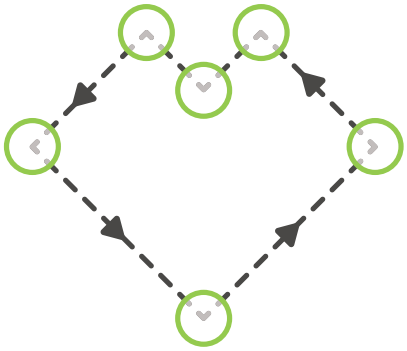
Lesson 5: Loopy Pictures

Connecting dots to make pictures is usually done with a pencil and paper. In this lesson, you'll make your pictures come to life by programming BOLT to connect the dots. Then you'll explore **loop controls** to make your programs repeat your code and go on and on and on…

Learning Objectives

1. I can use **loop forever** and **loop x times blocks** to execute sections of my program over and over again.
2. I can program BOLT to trace the outline of a picture I design.

Setup



Program QR Code

sphero.cc/BOLT5

Exploration:

Place paper Targets to create the outline of a familiar shape. Keep it simple by using eight or fewer Targets.

Skills Building:

Program BOLT to connect the dots and trace your outline with **roll** and **delay blocks**. Then explore what happens when you add a **loop forever block**.

Challenge:

Remix the program to make BOLT trace your outline **exactly three times**. *Which block did you need to use?*

Extended Challenge:

Ready for more? Try some of the following challenges:

- Explore the **loop until block**. Change the program so that BOLT stops after **time elapsed is greater than 20** or **distance is greater than 200**.
- Use the **matrix animation block** to change your BOLT shape into a light show.

Lesson 6: Word Games with BOLT

Overview

Have you ever played Mad Libs, the word substitution game? Maybe you've played it on a long car trip or at a party to share a few laughs with friends, but in this lesson, your students will play a similar game with BOLT. In the process, they'll review nouns, adjectives, and verbs, and get into some advanced block programming!

Programming Level

Advanced Block

Learning Objectives

1. I can define **string variables** as nouns, adjectives, and verbs.
2. I can understand how a program uses **functions** to keep code organised.
3. I can use the orientation sensor to affect how a program executes.

Vocabulary

- **Function**: Groups of blocks that can be called, or reused, throughout a program
- **Sensor**: Information gathered by BOLT including location, orientation, and acceleration
- **Yaw orientation**: The spin (twist) angle of BOLT from -180° to 180°

Reference to the Australian Curriculum: Digital Technologies Content descriptions

- Year 5-6 Implement digital solutions as simple visual programs involving branching, ~~*iteration (repetition)*~~, and user input (ACTDIP020)
- NOTE: This lesson does not address all aspects of this particular Content description, and these are indicated by the italicised wording above.

Content Connections

- language arts
- computer science

Materials

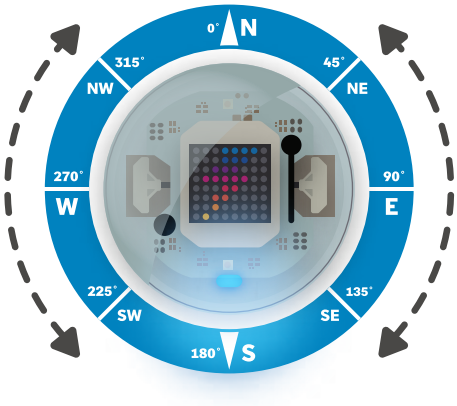
- BOLT Power Pack with robots fully charged
- Programming devices with the Sphero Edu app installed
- Protractors

Preparation

- This program does not require floor space. Students can use BOLT at their desks.
- Preview the program associated with this lesson to anticipate trouble spots for your students.
- Preview and prepare to share the student instructions (page 56).

Setup

Students will place BOLT inside of the Protractor and manipulate its yaw orientation.

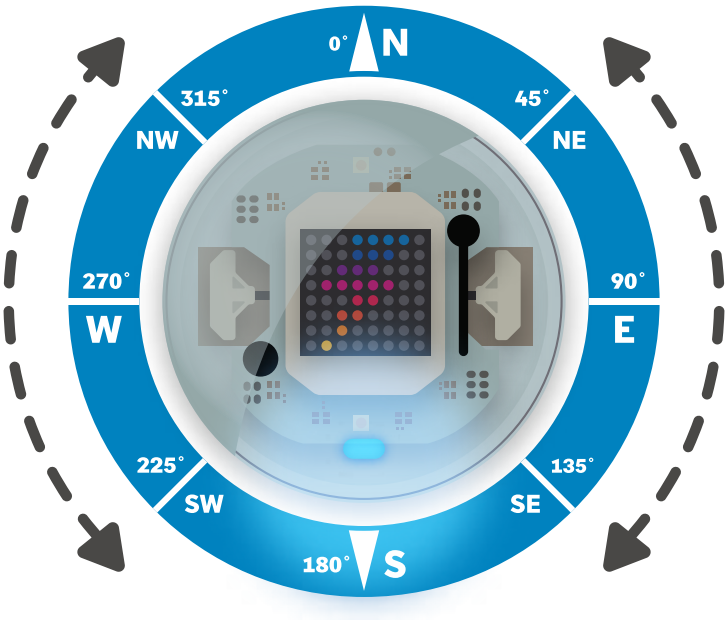


Program QR Code

sphero.cc/BOLT6

Exploration

1. Review the concept behind Mad Libs.
   - Mad Libs is a word game where you make silly stories by choosing nouns, verbs, adjectives, and other parts of speech before you know what the story is about.
   - If time allows, play a few printable versions of the games on the Mad Libs website (sphero.cc/bf11f8).
2. Explain that students will be playing a similar game with their BOLT robots.
3. Invite students to execute the program associated with this activity:
   - Pair BOLT with their programming devices and open the program.
   - Place BOLT on their desktops inside of the Protractor.
   - Aim the BOLT so that it is pointed directly at 0° North and the blue LED light is pointed 180 degrees South.
   - Tell students to execute the program multiple times and experiment with changing the orientation of BOLT.

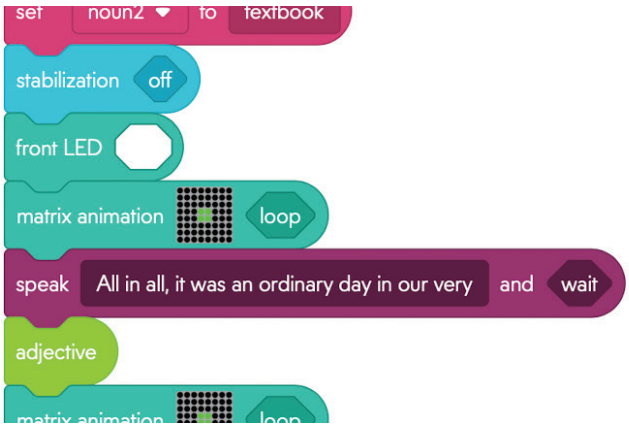


4. Discuss what happened. *Could you control which words the program substituted into the program? If so, how?*

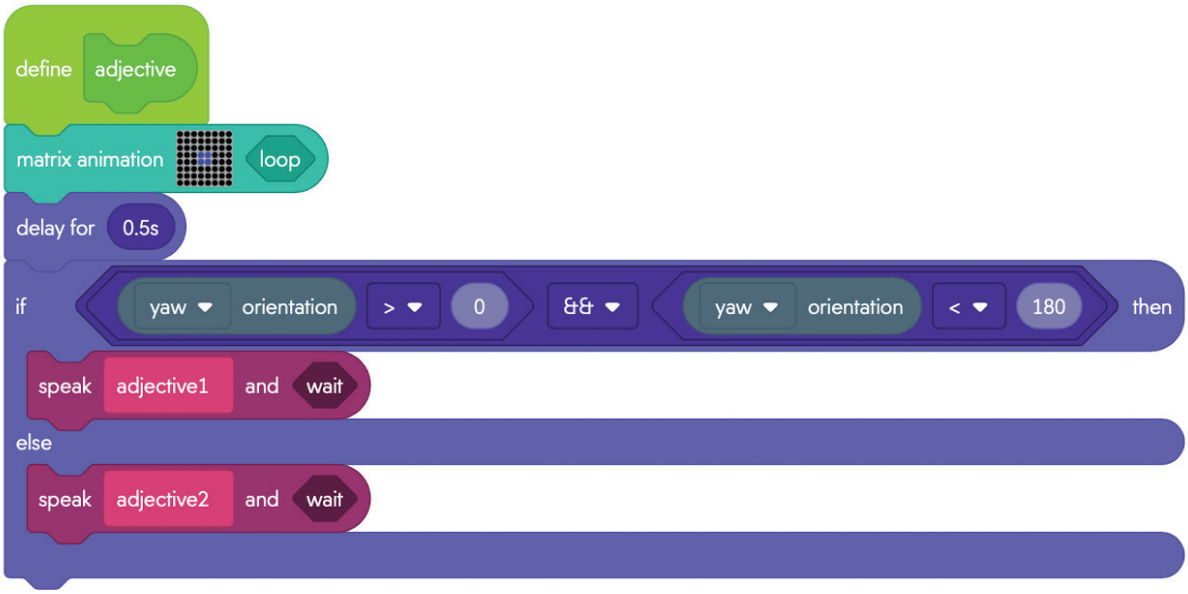
Skills Building

1. Ask students to analyse the blocks in the program for blocks that they recognise. They will likely recognise the **matrix animation** and **speak blocks**. Other blocks may be new to them.
2. Talk through parts of the program to show students how it works.
   - First, the program defines **six string variables**. These are the words that BOLT will substitute into the story.

   - Then, the program turns **stabilisation off** (so BOLT can rotate without the motor moving), lights up the **front LED** (so you can tell which way BOLT is heading), shows a **matrix animation**, and **speaks** the first part of the story.



   - The green block, **adjective**, is a **function**. A **function** is a tool that programmers use to save time and keep their code organised. Anytime a programmer wants to use the blocks attached to the **define adjective block**, they "call", or use, the **adjective block** in the main program. If the orientation sensor detects that BOLT is pointed between 0 and 180°, then it will read the value for **adjective1**. If BOLT is pointed in the other direction, it will read the value for **adjective2**.

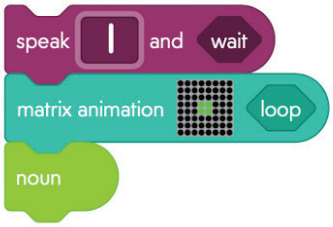


TEACHER TIP: Depending on your students prior programming experience, there may be a lot of new vocabulary and concepts in this lesson. However, the program works "as is." Adjust the depth of your explanation according to the level of your students.

3. Ask students to insert their own words into the **string variables** at the top of the program. Review the meaning of adjectives, verbs, and nouns as needed. Then challenge students to use the orientation of BOLT to control how the story is told.

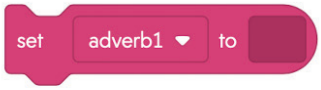
Challenge

1. Introduce the challenge: This story works, but it is a little short. Continue the programming pattern in the main program to continue the story.
   - Students need to add a green **matrix animation block**.
   - Then, they need to add a **speak block** with a new line for the story.
   - Finally, they need to add an **adjective, verb**, or **noun function**.



Extended Challenge

1. If your students have additional time, prompt them to try some of the following:
   - Change the text in the **speak blocks** to make a new Mad Lib story.
   - Add sound effects to make the story come to life.
   - Attach blocks to **define adverb function**. Place new **set variables blocks** at the top of the program and set values for **adverb1** and **adverb2**.



INTERESTED IN MORE LANGUAGE ARTS ACTIVITIES?

Try out some of the Sphero language arts activities listed on the STEAM Activities and Skill Progression Chart (sphero.cc/chart).

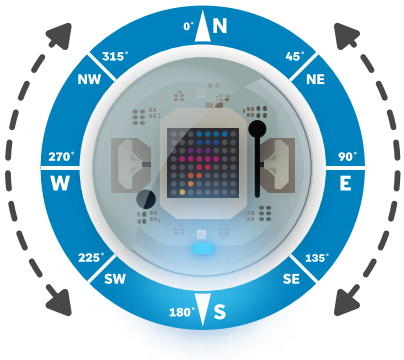
Lesson 6: Word Games with BOLT

Have you ever played Mad Libs, the word substitution game? Maybe you've played it on a long car trip or at a party to share a few laughs with friends, but in this lesson, you'll play a similar game with BOLT. In the process, you'll review nouns, adjectives, and verbs, and get into some advanced block programming!

Learning Objectives

1. I can define **string variables** as nouns, adjectives, and verbs.
2. I can understand how a program uses **functions** to keep code organised.
3. I can use the orientation sensor to affect how a program executes.

Setup



Program QR Code

sphero.cc/BOLT6

Exploration:

Place BOLT inside of the Protractor, open the program, and run it a few times. Can you spin BOLT to control which words it substitutes into the story?

Skills Building:

Analyse the blocks in the program to try to figure out how it works. Then, input your own words for **adjective1**, **adjective2**, **adjective3**, **adjective4**, **noun1**, and **noun2** at the top of the program to change up the story.

Challenge:

Continue the story. Add a **matrix animation block**, a **speak block** with a new line, and a green **adjective**, **verb**, or **noun function block**.

Extended Challenge:

Ready for more? Try programming:

- Change the text in the **speak blocks** to make a new Mad Lib story.
- Add sound effects to make the story come to life.
- Attach blocks to **define adverb function**. Place new set variables blocks at the top of the program and set values for **adverb1** and **adverb2**.

Lesson 7: BOLT Plays with Probability

Overview

Plinko is one of the most popular games on The Price is Right game show. In this lesson, your class will set up a similar game for BOLT and explore the probability of different outcomes. In the process, your students will learn more about **number variables** and **if else controls**.

Programming Level

Advanced Block

Learning Objectives

1. I can generate random numbers and use **if else controls** to affect BOLT's movement and behaviour.
2. I can read and leave comments in programs to communicate with other programmers.

Vocabulary

- **Integer**: Set of positive and negative whole numbers and 0
- **Pseudocode**: Informally written code that is simply meant to map out and describe actual code
- **Comment**: Notes that programmers leave in their code that explain their thinking; does not interfere with code

Reference to the Australian Curriculum: Digital Technologies Content descriptions

- Year 5-6 Implement digital solutions as simple visual programs involving branching, ~~*iteration (repetition)*~~, and user input (ACTDIP020)
- NOTE: This lesson does not address all aspects of this particular Content description, and these are indicated by the italicised wording above.

Content Connections

- computer science
- maths

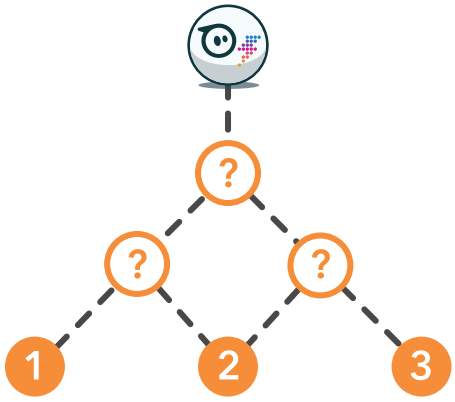
Materials

- BOLT Power Pack with robots fully charged
- Programming devices with Sphero Edu app installed
- Targets (page 71)

Preparation

- Plan for student work areas around the learning space. Each student group will need an approximately 5' by 5' area of floor space.
- Label and set up Targets to make one gameboard ahead of time and prepare to play the game with students to introduce the activity. Place Targets approximately two feet apart.
- Preview and prepare to share the student instructions (page 62).

Setup

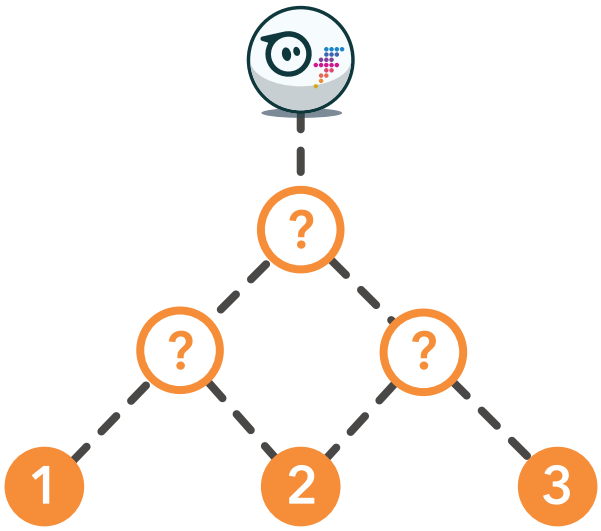


Program QR Code

sphero.cc/BOLT7

Exploration

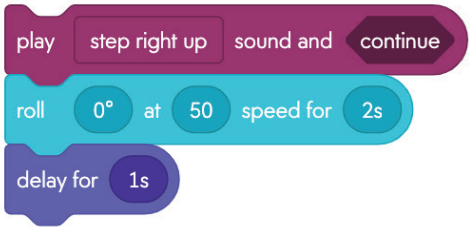
1. Introduce the concept behind Plinko. If possible, show a quick video to get students onboard, like this one of a high schooler winning the game (sphero.cc/b78ec1).
2. Explain that you'll be playing a version of Plinko with BOLT. The main difference is that instead of choosing a starting point, BOLT will always start in the middle.



3. Play a round with students:
   - Pair BOLT with a programming device and open the program associated with this lesson.
   - Place BOLT in its starting position and aim it towards the first Target.
   - Ask students where they think BOLT will "land."
   - Select Start to run the program.

Skills Building

1. Invite students to set up the game area with Targets in their work area and play a few rounds.
2. Prompt students to read the comments in the program to try to figure out how the program works. Explain that **comments** are notes that programmers leave in their code to explain their thinking.
3. Ask students to take turns explaining how the program works to their learning partners. Students may recognise:
   - BOLT first plays the step up sound and then rolls to the first Target.



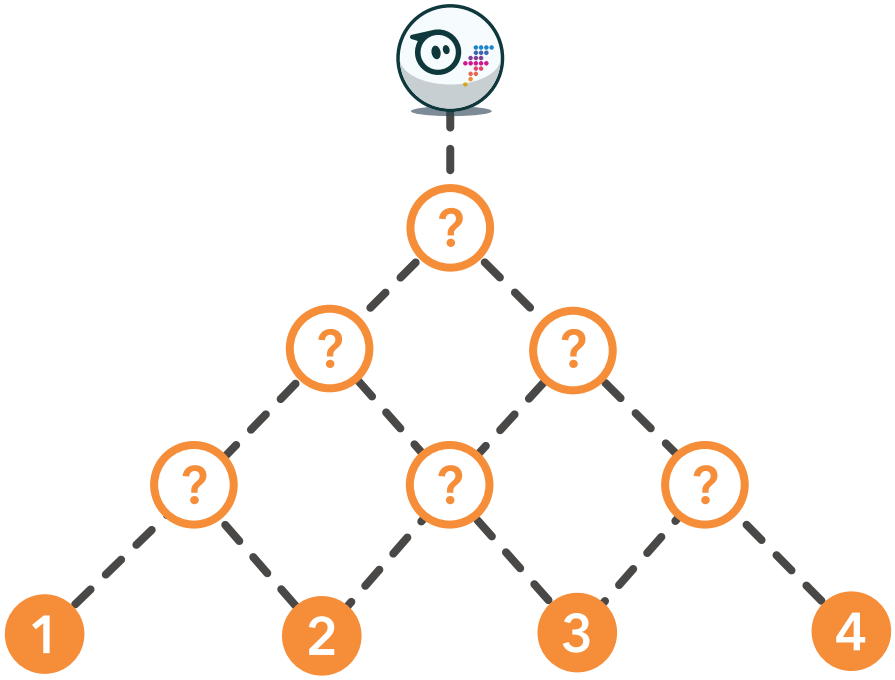
- Then the program chooses a random number, either 0 or 1, and sets a **variable** called **leftRight** to that number.

- The **if else control block** decides which way BOLT will roll. If **leftRight** is equal to 0, BOLT will roll at a heading of 45°. If l**eftRight is equal to 1**, BOLT will roll at a heading of 315°.

TEACHER TIP: Explain that students just explained the program in **pseudocode**, or informal language that describes how programs work.

Challenge

1. Ask students to extend the game area one more level.



2. Challenge students to extend the program so that BOLT continues to roll randomly to the new level.
   - Students can either drag blocks out to continue the program or they can long press or right click on a block to duplicate and reuse it.
   - Students need to add another **set leftRight block** and another **if else control block** to the end of the program.
3. Ask students to predict where BOLT will land most often. As time allows, turn this into an opportunity to compare the theoretical and experimental probabilities of different outcomes.
   - Outcomes #1 and #4 have only one possible path.
   - Outcomes #2 and #3 have three possible paths.

     Does this match the outcomes students find through experimentation?

Extended Challenge

1. To extend student learning in this lesson, consider asking your students to investigate one of the following options:
   - Add more rows to the game and look for patterns about the most likely outcome.
   - Look at the program and you'll notice that some blocks are repeated. Create a **function** called **decisionPoint**. Move the blocks that repeat to the **function** and then call the function in the main program.

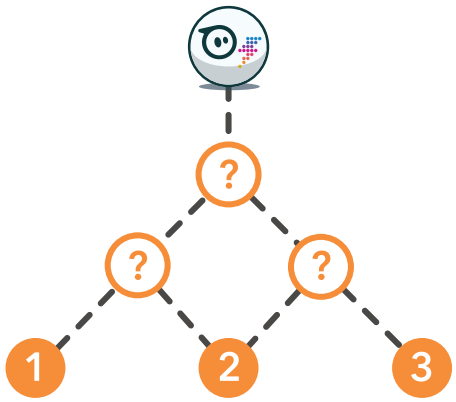
Lesson 7: BOLT Plays with Probability

Plinko is one of the most popular games on The Price is Right game show. In this lesson, set up a similar game for BOLT and explore the probability of different outcomes. In the process, you'll learn more about **number variables** and **if else controls**.

Learning Objectives

1. I can generate random numbers and use **if else controls** to affect BOLT's movement and behaviour.
2. I can read and leave comments in programs to communicate with other programmers.

Setup



Program QR Code

sphero.cc/BOLT7

Exploration:

Set up your game area. *Where do you think BOLT is most likely to land*?

Skills Building:

Open the program, aim BOLT at the first Target, and run the program. This program uses a number variable, called **leftRight**, and an **if else control block**. Talk to your learning partners about how they affect how BOLT moves in the program.

Challenge:

Add another row to the game area and then add more blocks to the program to make BOLT make another choice and roll a third time!

Extended Challenge:

Ready for more? Try some of the following challenges:

- Add more rows to the game area and look for patterns about the most likely outcome.
- Look at the program and you'll notice that some blocks are repeated. Create a **function** called **decisionPoint**. Move the blocks that repeat to the **function** and then call the **function** in the main program.

Lesson 8: Roll BOLT at the Sphero Arcade

Overview

Test your programming skills and "roll" BOLT to see how many points you can score at the Sphero Arcade. In this lesson, your students will code their first programs in JavaScript. Then, they'll cover their BOLTs with Turbo Covers and explore how to adjust their programs to match the new material.

Programming Level

Beginning Text

Learning Objectives

1. I can program BOLT with JavaScript.
2. I can describe how the Turbo Cover affects BOLT's movement and adjust my program accordingly.

Vocabulary

- **JavaScript**: The text programming language that runs Sphero's block programming environment
- **Text Canvas**: The editor in the Sphero Edu app for creating JavaScript programs
- **Statement**: Programming instructions in JavaScript
- **Debug**: Locate and troubleshoot errors in a program

Reference to the Australian Curriculum: Digital Technologies Content descriptions

- Year 7-8 Implement and modify programs with user interfaces ~~*involving branching, iteration and functions*~~ in a general- purpose programming language (ACTDIP030).
- NOTE: This lesson does not address all aspects of this particular Content description, and these are indicated by the italicised wording above.

Content Connections

- computer science

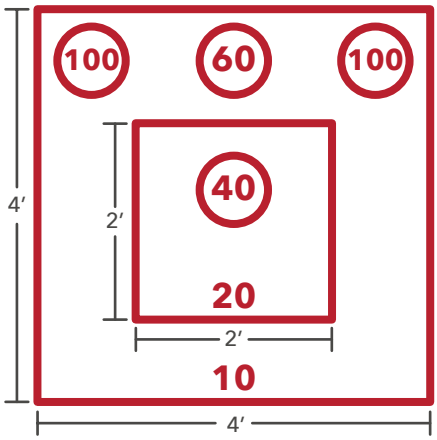
Materials

- BOLT Power Pack with robots fully charged
- Programming devices with Sphero Edu app installed
- Maze Tape (included with BOLT Power Pack)
- Targets (page 71)
- Sphero Covers (included with BOLT Power Pack)

Preparation

- Set up 3-4 game stations around the room. Label Targets to make numbers and use Maze Tape to outline the squares.
- Plan for 3-4 teams at each station.
- Review the Getting Started section of the Sphero Programming Wiki to learn more about programming BOLT with JavaScript (sphero.cc/JS-help).
- Preview and prepare to share the student instructions (page 67).

Setup

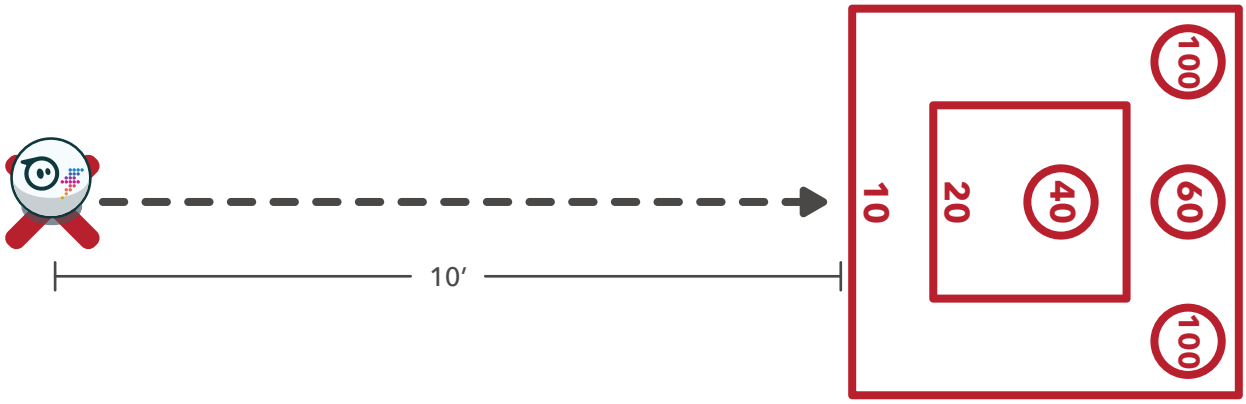


Program QR Code

sphero.cc/BOLT8

Exploration

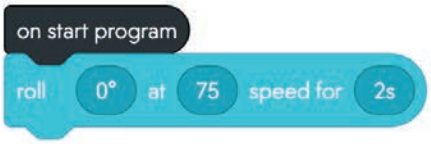
1. Tell students that they are going to play an arcade-style game with Sphero BOLT.
2. Show students one of the game setups around the classroom and explain that students will "roll" BOLT to see how many points they can score in the target area.



3. As time allows, give students a few minutes to play the game by programming BOLT on the Block Canvas.
   - Place BOLT 10 feet in front of the scoring area.
   - Aim the robot at the desired score.
   - Take turns programming BOLT with the **roll block**.

Skills Building

1. Tell students that in order to make this game more challenging, they will program BOLT to roll to Targets with JavaScript.
   - JavaScript is one of the world's more popular programming languages.
   - JavaScript is the programming language behind the Block Canvas that students are already familiar with.
2. Ask students to compare and contrast the **roll block** and the roll statement. *How are they the same? How are they different?*



Blocks

```
1 async function startProgram() {
2     await roll (0, 75, 2);
3 }
4
```

JavaScript

- They both have the same three inputs: heading, speed, and duration.
- The block says "roll," while the JavaScript reads, "await roll."

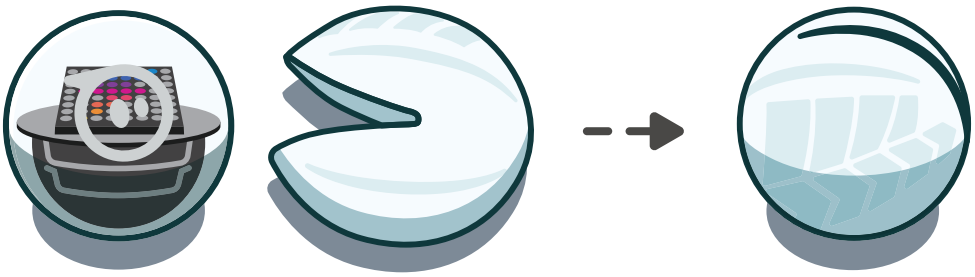
TEACHER TIP: The word "async" at the start of the program indicates that this is an asynchronous program. The word "await" at the start of the roll statement indicates that this statement must be fully executed before the program moves on to the next line.

3. Explain that JavaScript has specific syntax or rules. Ask students to look for the following syntax in the `await roll()` statement.
   - ; ends a statement
   - ( ) contains a value
   - , separates values
4. Ask students to open the Text Canvas associated with this activity and use the `await roll (0, 0, 0)` command to play a few rounds of the game.
   - *Which inputs reliably score the most points?*
   - *Is BOLT's movement more precise with the slower speeds and longer durations or faster speeds and shorter durations?*

TEACHER TIP: If a student types a statement incorrectly, the program may not execute. These types of syntax errors are not possible to make on the Block Canvas. Tell students that they will have to **debug**, or find and troubleshoot their errors, to get their program to work.

Challenge

1. Ask students to place the Turbo Cover over their robots.



2. Challenge students to remix their programs to play the game with Turbo Covers on their robots.
3. Invite students to play a few rounds of the game versus their classmates.
4. Discuss with students:
   - *How do covers affect BOLT's movement?*
   - *What did you need to adjust in your program to maintain your robot's accuracy?*

Extended Challenge

1. Once your students have coded their first text program, they'll be ready for more. Suggest some of the following challenges:
   - Use the Targets to design a mini golf course. Make holes that require more than one `await roll()` statement.
   - Try some other BOLT JavaScript statements like

     ```
     await spin(360, 1)
     setMainLed({ r: 90, g: 255, b: 90 })
     await Sound.Animal.play(true)
     ```

     What do they do?
   - Explore the Sphero Programming Wiki (sphero.cc/JS-help) to learn more about JavaScript programming.

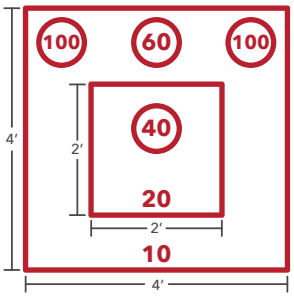
Lesson 8: Roll BOLT at the Sphero Arcade

Test your programming skills and "roll" BOLT to see how many points you can score at the Sphero Arcade. In this lesson, you'll code your first program in JavaScript. Then, you'll cover your BOLT robot with a Turbo Cover and explore how to adjust your programs to match.

Learning Objectives

1. I can program BOLT with JavaScript.
2. I can describe how the Turbo Cover affects BOLT's movement and adjust my program accordingly.

Setup



Program QR Code

sphero.cc/BOLT8

Exploration:

Play a few rounds of the game by programing BOLT to roll on the Block Canvas

- Place BOLT 10 feet in front of the scoring area.
- Aim the robot at the desired score.
- Take turns programming BOLT with roll blocks.

Skills Building:

Open the Text Canvas and program BOLT to roll to the scoring area with the `await roll()` statement.

Challenge:

Add the Turbo Cover to your BOLT and play some more. The Turbo Cover adds traction but you may have to adjust your code!

Extended Challenge:

Ready for more? Try programming:

- Use the Targets to design a mini golf course. Make holes that require more than one `await roll()` statement.
- Try some other BOLT JavaScript statements like

```
await spin(360, 1)
setMainLed({ r: 90, g: 255, b: 90 })
await Sound.Animal.play(true).
```

  What do they do?
- Explore the Sphero Programming Wiki (sphero.cc/JS-help) to learn more about JavaScript programming.

Go Further

After your students have completed some or all of the classroom activities in this guide. They'll be ready to continue learning with other activities in the Sphero Edu app, both created by Sphero or by our community of educators.

To continue teaching the fundamentals of programming, consider the following activities appropriate to the level of your students.

INTRO TO DRAW

INTRO TO BLOCKS

INTRO TO TEXT

For programming activities designed specifically to showcase the features of BOLT, try the following sequence of activities.

1. Meet Sphero BOLT
2. BOLT: Maze
3. BOLT: Musical Form
4. BOLT: Light Sensor
5. BOLT: Matrix
6. BOLT: Infrared
7. BOLT: Retro Games
8. BOLT: Compass

Open the chart, to explore more BOLT activities for teaching content across subject areas and grade bands.

sphero.cc/chart

Additional Curricular Offerings

Sphero Computer Science Foundations

Computer Science Foundations is a dynamic, standards-aligned curriculum designed to be taught in the classroom with BOLT robots. Learn computer science and STEAM principles over three courses, with 72 sequenced lessons allowing teachers and students to learn and grow together. If you've been looking to integrate computer science content into your STEAM classroom, look no further.

Sphero Code Mat

A Code Mat offers a simple, accessible way to learn block-based coding, basic maths principles, and collaborative problem-solving with your class set of BOLT robots.

Each two-sided Code Mat has a matching set of 3 x 10 double-sided coding cards that provide guided, hands-on coding lessons (sold separately). Choose either the Sphero City and Golf Code-themed Mat and cards or the Sphero Space and Soccer Code-themed Mat and cards to energise your Sphero classroom.

For more information on Sphero, head to www.sphero.com.au

Resources

Sphero is inspiring the creators of tomorrow. We couldn't be more excited about the future of education and the part we're all playing. For more information about Sphero and to get involved in our community you can find links to additional resources below.

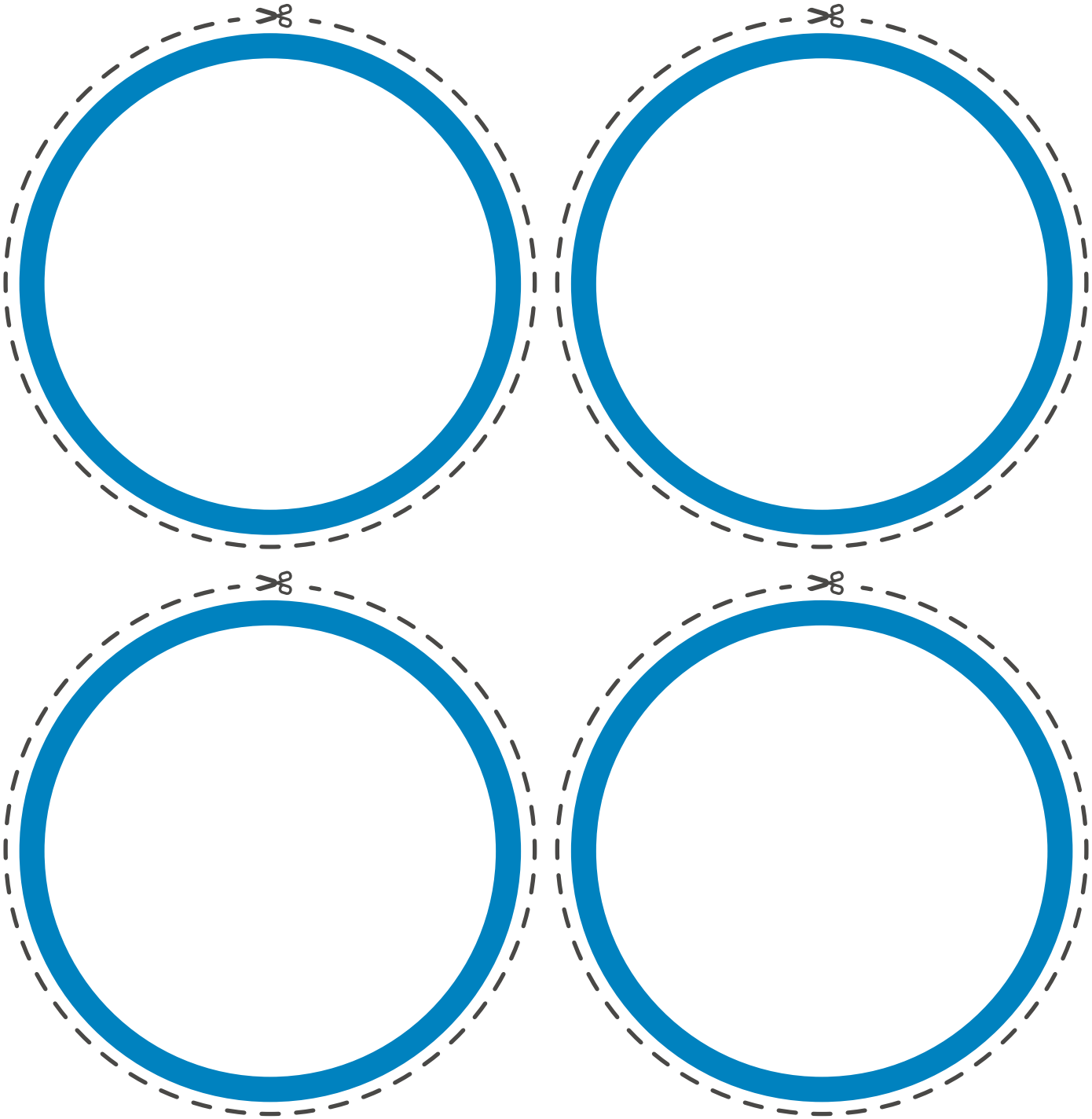
- Sphero Blog: sphero.com/blogs/news
- Sphero Programming Wiki: sphero.cc/JS-help
- Support: support.sphero.com
- Self-service resources: sphero.com/pages/sphero-self-service-resources
- Contact Us: sphero.com/pages/contact-us
- Facebook: facebook.com/GoSphero
- Twitter: twitter.com/sphero
- Instagram: instagram.com/sphero

Security & Privacy

We are dedicated to ensuring BOLT and the Sphero Edu app is safe and secure to use. Our security and privacy efforts include third party testing, annual audits, and a bug bounty program. We are COPPA compliant, have signed the Student Privacy Pledge, and publish all of our privacy practice agreements online. Visit https://www.sphero.com/privacy for more information.

Targets

Make copies of this page as needed for guidebook activities. Cut targets along the dotted lines and label them per the individual activity instructions. Access a digital version for printing at sphero.cc/targets.



Glossary

Draw Canvas: The editor in the Sphero Edu app for creating programs by drawing lines (Lesson 1)

Execute: To run or complete a block or program (Lesson 1)

Block Canvas: The editor in the Sphero Edu app for creating block programs (Lesson 2)

Input: Place to add numeric or text values into a programming block or text statement (Lesson 2)

Speed: Input to program how quickly (or slowly) a BOLT will move (Lesson 2)

Heading: Input to program the direction a BOLT is pointed during or before a roll, between 0° and 360° (Lesson 2)

Duration: Input to program the amount of time that BOLT will execute a movement like a roll or a spin in seconds (Lesson 2)

Asynchronous: Programming that executes multiple blocks or commands at the same time (Lesson 3)

Synchronous: Programming that executes one block or command at a time (Lesson 3)

Variable: A piece of information, like a number value or string, that can change during a program (Lesson 4)

String: Phrases that combine letters, numbers, and/or other characters (Lesson 4)

Control: A programming tool used to make certain parts of a program repeat or execute under certain conditions (Lesson 5)

Loop: A programming control that makes an action of repeating over and over again (Lesson 5)

Function: Groups of blocks that can be called, or reused, throughout a program (Lesson 6)

Sensor: Information gathered by BOLT including location, orientation, and acceleration (Lesson 6)

Yaw orientation: The spin (twist) angle of BOLT from -180° to 180° (Lesson 6)

Integer: Set of positive and negative whole numbers and 0 (Lesson 7)

Pseudocode: Informally written code that is simply meant to map out and describe actual code (Lesson 7)

Comment: Notes that programmers leave in their code that explain their thinking; does not interfere with code (Lesson 7)

JavaScript: The text programming language that runs Sphero's block programming environment (Lesson 8)

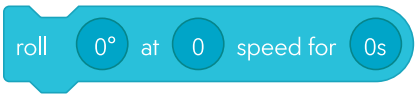
Text Canvas: The editor in the Sphero Edu app for creating JavaScript programs (Lesson 8)

Statement: Programming instructions in JavaScript (Lesson 8)

Debug: Located and troubleshoot errors in a program (Lesson 8)

Block Library

Movements



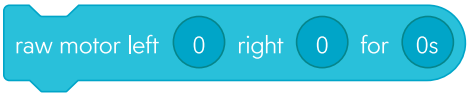
Roll: Combines heading, speed, and duration to make the robot roll.

Stop: Sets the speed to 0 to stop the robot when a Speed block is used.

Speed: Sets the target speed of the robot from -255 to 255. Positive speed is forward, negative speed is backward, and 0 is stopped.

Heading: Sets the target direction the robot rolls. Assuming the robot is aimed with the blue tail light facing you, then 0° is forward, 90° is right, 270° is left, and 180° is back.

Spin: Spins the robot for a given number of degrees over time, where 360° is a single rotation.



Raw Motor: This command disables the stabilisation and will cause the robot to jump when both motors are set to full power. Controls the electrical power sent to the left and right motors independently, from -255 to 255, for a duration of seconds.

**BOLT only*

Stabilisation: Turns the stabilisation system ON or OFF. Stabilisation is ON by default to keep the robot upright.

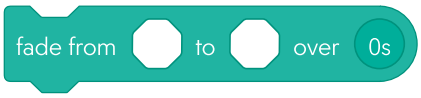
Calibrate Compass: Calibrates the magnetometer by spinning the robot in place. You need to run this command before setting or getting the compass direction. Nearby metallic and magnetic objects can affect the accuracy of the compass.

Compass Direction: Sets the real-world direction based on the last compass calibration. 0° is due north, 90° is due east, 180° is due south, and 270° is due west. Requires the Calibrate Compass command to be run before you can set this value. Nearby metallic and magnetic objects can affect the accuracy of the compass.

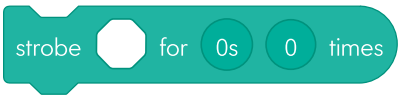
Reset Aim: Resets the heading calibration (aim) angle to use the current front-facing direction of the robot as 0°.

Lights

Main LED: Changes the colour of the main LEDs. Set this using the colour wheel and brightness slider, or the exact RGB (red, green, blue) values, from 0 to 255.



Fade: Changes the main LEDs from one colour to another, for a duration of seconds. This command works as a standalone command and cannot be used to fade main LEDs while using a roll command.

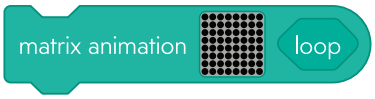


Strobe: Blinks the main LEDs for a period of seconds (that includes lights ON and OFF) and a count of cycles. A short period will produce a fast blink whereas a long period will produce a slow blink.

Back LED: Changes the colour of the back LED. Turns on the back LED. This is limited to blue only. You can adjust the brightness by tapping on the number.

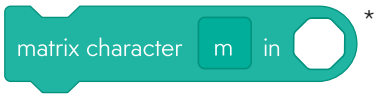
*

Front LED: Changes the colour of the front LED. Set this using the colour wheel and brightness slider, or the exact RGB (red, green, blue) values, from 0 to 255.



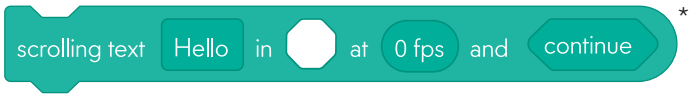
*

Matrix Animation: Sets an image or animation (> 1 frame) on the 8x8 LED matrix, limited to 16 colours per animation. You can modify the frame speed from 1 to 30, and a transition style to fade, or not fade, between frames. Animations will play until interrupted by a new animation or pause/ clear command. Create your own or use some of Sphero's pre-made animations.



*

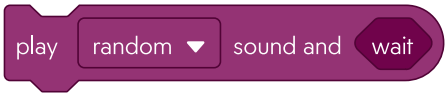
Matrix Character: Displays a single ASCII character on the LED matrix, in a specified colour.



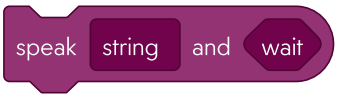
*

Matrix Scroll Text: Display a string of characters, or use the string builder to include or concatenate variables and sensor values. You are limited to 26 ASCII characters, a single text colour, and can set a frame speed from 1 to 30.

Sounds



Sound: Plays a sound from your programming device (not from the robot) that you select from the list. Toggle the Randomize option to generate a random sound or select one from Sphero's library.

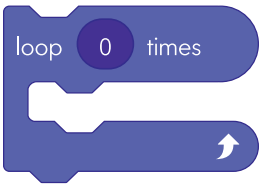


Speak: Speak strings (numbers and words) from your programming device. Type what you want your robot to say.

**BOLT only*

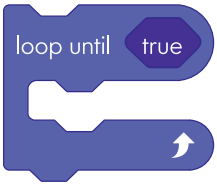
Controls

Delay: Delays execution of the next block for a number of seconds.



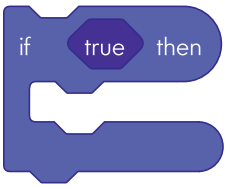
Loop: Repeats the blocks within for the number of loops specified, also known as a "for" loop.

Loop Forever: Repeats the blocks within forever, also known as a "while 1" loop. You can use this to constantly evaluate conditions.

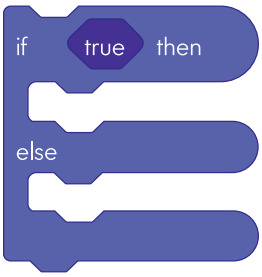


Loop Until: Repeats the blocks within until a condition is met, also known as a "while" loop. Drag any comparator into the "true" field to create a condition. Use "and" and "or" to combine or exclude conditions, which must be added before other comparators.

Exit Program: Stops all code and ends the program; the same as hitting the Stop button.

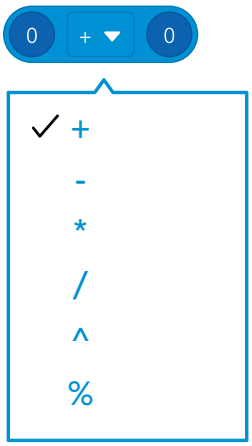


If Then: Calls the "if" blocks if the given condition is true. Use "and" and "or" to combine or exclude conditions, which must be added before other comparators.



If Then Else: Calls the "if" blocks if the given condition is true; otherwise calls the "else" blocks. Use "and" and "or" to combine or exclude conditions, which must be added before other comparators.

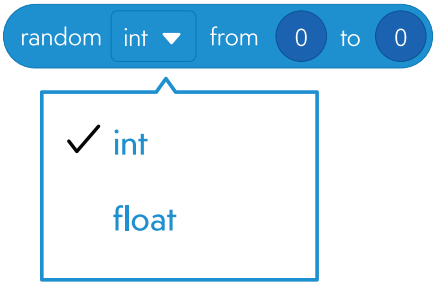
Operators



Basic Operators: Adds a mathsematical action to calculate or evaluate a set of data or given value.

**BOLT only*

Build String: Combines multiple values into a single string. Those values can be numbers (variables, parameters, or sensors), strings, booleans, or colours.



Random Int: Generates a random integer value (nearest whole number) within the given minimum and maximum.

Random Float: Generates a random float value (including decimals) within the given minimum and maximum.

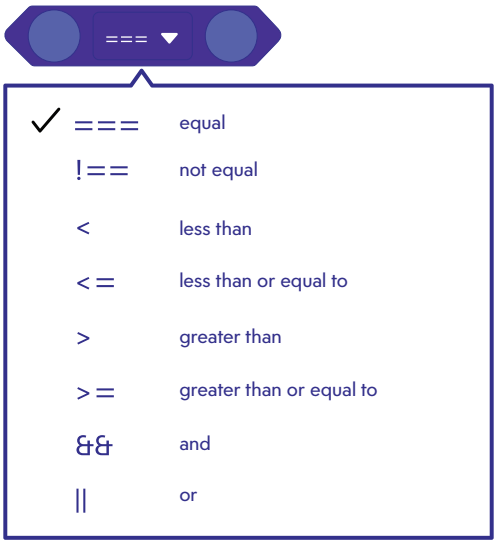
Colour Channel: Gets the value of the red, green, or blue colour channel for a given RGB value.

Colour Mixer: Returns a new colour by modifying a single channel (red, green, blue) of a given colour.

Random Colour: Sets a random colour when placed in a colour block.

**BOLT only*

Comparators



Comparators: Comparators are blocks of code that compare two things (Sensor data, operators, strings, etc.). Comparators allow us to create logical checks, which, when "true," will allow the program to perform a certain action or event. If a comparison is true and logical, then the code will be executed.

Sensors

Accelerometer - Total: The accelerometer sensor measures the change in force in the robot.

- Total: The combined vector acceleration of all three axes, from of 0 to 14 g's.
- X-Axis: The left-to-right acceleration, from -8 to 8 g's.
- Y-Axis: The forward-to-back acceleration, from -8 to 8 g's.
- Z-Axis: The up-and-down acceleration, from -8 to 8 g's.

pitch ▼ orientation

Orientation: The orientation sensor measures the orientation of the robot.

- Pitch: The forward or backward tilt angle, from -180° to 180°.
- Roll: The left or right tilt angle, from -90° to 90°.
- Yaw: The spin (twist) angle, from -180° to 180°.

pitch ▼ gyroscope

Gyroscope: The gyroscope sensor measures the rate of spin in the robot in degrees per second.

- Pitch: The rate of forward or backward spin, from -2,000° to 2,000° per second.
- Roll: The rate of left or right spin, from -2,000° to 2,000° per second.
- Yaw: The rate of sideways spin, from -2,000° to 2,000° per second.

total ▼ velocity

Velocity - Total: The velocity sensor measures the speed in centimetres per second.

- Velocity Total: The combined vector speed of both axes which will always be a positive value, in centimetres per second.
- Velocity - X: The right (+) or left (-) speed, in centimetres per second.
- Velocity - Y: The forward (+) or back (-) speed, in centimetres per second.

total ▼ location

Location - Total: The distance from the location of the program start, which will always be a positive value, in centimetres.

**BOLT only*

distance

Distance: The total distance traveled, in centimetres.

speed

Speed: Target speed value, from -255 to 255.

heading

Heading: Target directional angle of the robot. Assuming you aim the robot with the blue tail light facing you, then 0° heading is forward, 90° is right, 180° is backward, and 270° is left.

main LED

Main LED: The RGB colour of the main LEDs, from 0 to 255 for each colour channel.

time elapsed

Time Elapsed: The amount of time with which the program has run (in seconds).

compass direction *

Compass Direction: The real-world offset between the aim heading and the last compass north reading.

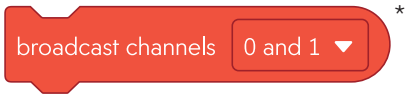
luminosity *

Luminosity: The light intensity from 0 - 100,000 lux, where 0 lux is full darkness and 30,000-100,000 lux is direct sunlight.

last message recieved *

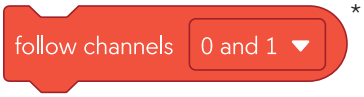
Luminosity: Returns which channel the last infrared message was received on.

Communications



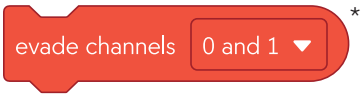
Broadcast: Sets the IR emitters to broadcast on two specified channels, from 0 to 7. The broadcaster uses two channels so the following or evading BOLTs can detect these messages on their IR receivers with a sense of relative proximity to the broadcaster. You can't use a channel for more than one purpose at a time.

Stop Broadcast: Stops the broadcasting behaviour.



Follow: Sets the IR receivers to look for broadcasting BOLTs on the same channel pair, from 0 to 7. Upon receiving messages from a broadcasting BOLT, the follower will adjust its heading and speed to follow the broadcaster.

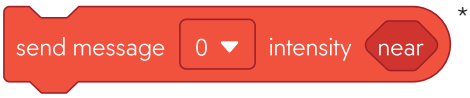
Stop Following: Stops the following behaviour.



Evade: Sets the IR receivers to look for broadcasting BOLTs on the same channel pair, from 0 to 7. Upon receiving messages from a broadcasting BOLT, the evader will adjust its heading to roll away from the broadcaster.

Stop Evading: Stops the evading behaviour.

**BOLT only*



Send Messaging: Sends a message on a given IR channel, from 0 to 7, at a set intensity, from one to 64. Intensity is proportional to proximity, where one is the closest, and 64 is the farthest. You can't overlap sending messages with broadcasting, following, or evading behaviours on the same channels.

Events

On Collision: Conditional logic called when the robot collides with an object.

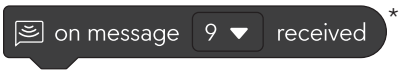
On Landing: Conditional logic called when the robot lands after an onFreefall event. You do not need to define an onFreefall event for the robot to experience an onLanding event, but the robot must meet the conditions for free fall before landing.

On Freefall: Conditional logic called when gravity is the only force acting on the robot, which is achieved by dropping or throwing it.

On Gyro Max: Conditional logic called when the robot exceeds the bounds of the measurable rotational velocity of -2000° to 2000° per second.

On Charging: Conditional Logic called when the robot starts charging its battery. This can be triggered by placing the robot in its charging cradle when the cradle is plugged in (Mini Excluded).

On Not Charging: Conditional logic called when the robot stops charging its battery. This can be triggered by removing the robot from its charging cradle or by unplugging it (Mini Excluded).



On Message Received: Conditional logic called when an infrared message is received on the specified channel.

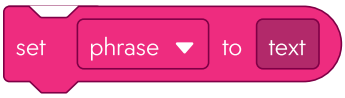
Variables

Number Variable: A stored number value that can be assigned and operated on. Values can be integers (whole numbers) or floating point values (numbers with decimals).

Set Number: Sets the number value for the selected number variable.

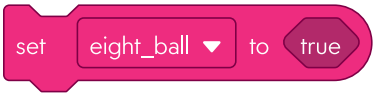
String Variable: A stored string value that can be assigned and operated on. Values can be words or sentences, such as "Hello World!"

**BOLT only*



Set String: Sets the string value for the selected string variable.

Boolean Variable: A stored boolean value that can be assigned and operated on. Values can be true or false.



Set Boolean: Sets the boolean value for the selected boolean variable.

Colour Variable: A stored colour value that can be assigned and operated on. Values are made up of Red, Green and Blue channels that range from 0 to 255.

Set Colour: Sets the colour value for the selected colour variable.

Functions

Function: Functions allow you to define a reusable group of blocks, and then call those blocks from anywhere within the main program. Parameters can be used like local variables to the function to change the behaviour each time the function is called.

sphero®

BOLT™

8 17961 02510 8

ISBN 978-1-7331447-5-9

90000>

9 781733 144759