



Sphero Edu

TEACHER RESOURCE GUIDE

YEAR 1-10 - Australian Classroom Edition

Proudly distributed by MacGear Australia | 02 9435 0387 | salesau@macgeargroup.com

Table of Contents

Introduction to Sphero Edu	3
Why is Sphero Edu a Valuable Educational Tool	6
Get to Know Your Robot	13
The Sphero Edu App	16
Sphero Edu Activities	18
Sphero Edu Units of Learning.	22
Supplemental Resources	32

Introduction to Sphero Edu

Sphero Edu empowers Australian teachers to foster curiosity and grow as thinkers and lifelong learners. Sphero Edu is a powerful toolset that supports learners and teachers to reinvent education. Our program not only supports progress with the Digital Technologies curriculum, but it also integrates all learning areas through STEAM activities. Through collaborative and innovative projects, learners will also develop their Capabilities and 21st Century thinking skills.

So how can Sphero Edu be used to address the Australian Digital Technologies Curriculum?

Although the possibilities are endless, below are some ideas of how you could integrate Sphero into your unique setting:

- Use computational thinking to address the Processes and Production skills strands of the Digital Technologies Curriculum by programming Sphero in three different ways designed for learner progression - draw, blocks, and JavaScript.
- Measure time, speed, distance and other metrics to support numeracy and science.
- Perform specific functions such as motion and direction, colour and light and sensor controlled reactions.
- Direct the movement of the Sphero using an app or autonomously using code to navigate a maze or terrain.
- Develop Key Capabilities by working collaboratively on authentic challenges.
- Use Design Thinking to solve and overcome challenges.

Sphero is adaptable to all ages, skill levels, and content areas. The Teacher Resource Guide is designed to give you everything you need to know to get the ball rolling with your learners.

Mission in Education

Sphero Edu provides a toolset that is unbounded in its potential. While some fundamental coding skills are necessary, our program also goes beyond code by incorporating robotics and technology with collaborative STEAM activities, nurturing students' imaginations in ways unique to our education program.

Sphero Robots in Education

Sphero Edu offers three different coding "canvases" - Draw, Block, and Text - that move from beginner to advanced coding skills, making it simple to use with learners of all ability levels, from primary to secondary school and special education. Students can work collaboratively or at their own pace thanks to the Sphero Edu app.

Sphero can be used in and out of the classroom in formal and informal learning environments. The following are some ways Sphero is being used in different learning environments.

Introduction to Sphero Edu



Primary School

Across Australia, Sphero is being used to teach computational thinking in Years 1-6. Learners are using Sphero Edu to explore all learning areas of the Australian curriculum, often through STEAM based activities. For Years 1-2, Sphero has empowered teachers to incorporate computational thinking with Play-Based learning strategies.

In Years 3-6, Sphero is allowing teachers to seamlessly integrate the new Digital Technologies curriculum with inquiry-based learning. This approach is providing opportunities for collaboration, critical thinking and increased engagement. Sphero's coding options allow computational thinking to be scaffolded and differentiated to suit students' skills. The intuitive interface and robustness of the robot also allow students to work autonomously with Sphero. This increases the development of student agency and empowers students to take risks.

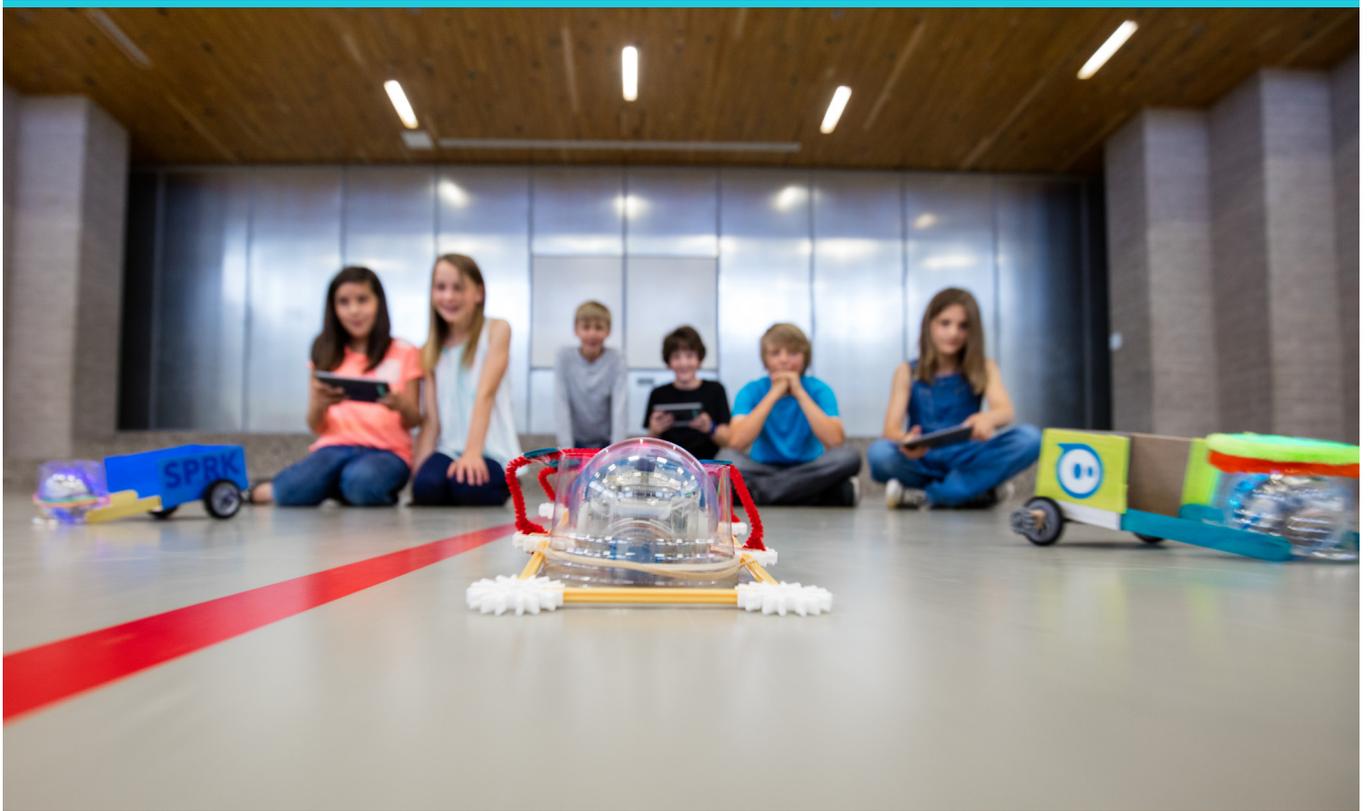
Upper Primary/Lower Secondary

Upper Primary/Lower Secondary school students are using Sphero to add depth to their learning and to also explore the Digital Technologies curriculum. As students develop their capabilities they begin to take on greater challenges which really test their collaboration and communication skills. Teachers are using Google, OneNote, Microsoft Teams and Apple Classroom to assign complex challenges to students and to track their progress with increasingly advanced computational thinking and problem solving strategies. As middle schools often have a diverse range of student needs, the flexibility of Sphero Edu allows all students to realise success no matter what their prior knowledge of computational thinking is.

Secondary School

Sphero is used for increasingly complex learning in Years 7-10. In years 7 and 8 students are using Sphero to add depth to Mathematics classes by exploring advanced concepts of logic, design thinking and computational thinking. They are using a variety of methods to simulate concepts and bring learning to life. In years 9 and 10, students are further developing their digital technologies knowledge while also using exportable data from the Sphero to analyse data from physics experiments and simulations. Throughout secondary school, students are engaging with the foundations of JavaScript and their programming skills are becoming increasingly advanced.

Introduction to Sphero Edu



Higher Education

In higher education and post-secondary learning environments, learners are using Sphero to provide a hands-on method to learn the programming language JavaScript and advanced coding skills. Sphero's built-in sensors are also used to measure forces and gather data, such as acceleration, velocity, and pitch, during scientific experiments.

Clubs & Competitions

Sphero is being used in robotics, coding, and STEM clubs that may meet before/after school or during lunch. Clubs can be an effective way to attract students, especially underrepresented groups, to participate in STEAM-based learning activities and creative challenges.

The hands-on nature of Sphero makes it the perfect tool for competitions. Competitions are a great way to bring younger and older students together. Kids can complete engineering challenges, program robots, and compete against peers. Think beyond formal competitions and organise a competition for your school such as Sphero Olympics, STEAM Night, or Evening of Code.

Makerspaces

Makerspaces are being implemented in schools everywhere, especially library programs, to encourage creativity, innovation, and hands-on learning. Sphero is the perfect addition to makerspaces, giving students an opportunity to learn by doing, tinker with robotics, and experiment with open-ended programming challenges. Refer to the [Makerspace Guide](#) for tips on bringing Sphero into a makerspace.

Why is Sphero Edu a Valuable Educational Tool?



Learning in the Technologies

Sphero is an incredibly versatile educational tool making it an ideal fit for STEAM, project-based learning, personalised learning and more.

Students need to be skilled in technologies and to be aware of the impact of technologies on society and the environment, now and in the future. Sphero's framework offers challenging activities which involve research, problem-solving, exploration of new and unfamiliar concepts, skills and materials, and the rewarding learning which often results from creating products which have real applications. Sphero allows for progression in cognitive skills. Students will develop their creativity and computational skills and be encouraged to become innovative and critical designers of the future.

Why is Sphero Edu a Valuable Educational Tool?



Purposes of Learning in the Technologies

Learning in the technologies enables children and young people to be informed, skilled, thoughtful, adaptable and enterprising citizens, and to:

- Develop understanding of the role and impact of technologies in changing and influencing societies
- Contribute to building a better world by taking responsible ethical actions to improve their lives, the lives of others and the environment
- Gain the skills and confidence to embrace and use technologies now and in the future, at home, at work and in the wider community
- Become informed consumers and producers who have an appreciation of the merits and impacts of products and services
- Be capable of making reasoned choices relating to the environment, to sustainable development and to ethical, economic and cultural issues
- Gain the skills and confidence to embrace and use technologies now and in the future, at home, at work and in the wider community
- Become informed consumers and producers who have an appreciation of the merits and impacts of products and services
- Be capable of making reasoned choices relating to the environment, to sustainable development and to ethical, economic and cultural issues

Why is Sphero Edu a Valuable Educational Tool?

Skills Developed in the Technologies

The technologies provide frequent opportunities for active learning in creative and work related contexts. Learning in the technologies thus provides opportunities to continually develop, use and extend skills that are essential components for life, work and learning, now and in the future, including planning and organisational skills. Learning in the technologies therefore makes a strong contribution to achieving the aim clearly articulated in the AU Digital Technologies Curriculum Progressions.

Educational Value and Alignment to Outcomes

Sphero can be used to teach mathematics, technology and supplement content in any learning area. Sphero provides extensive learning activities that are aligned to the Digital Technologies curriculum and teachers can tailor activities towards the principles of the Australia Curriculum by integrating Information and Communication Technology, Numeracy, Literacy, Inter-Cultural Understanding, Personal and Social Capability, Critical and Creative Thinking and Ethical capabilities into each project.

Teachers don't have to be programming experts to integrate Sphero into their classroom instruction. Sphero Edu offers three different coding "canvases" - Draw, Block, and Text - that move from beginner to advanced coding skills. The three coding canvases make it easy for teachers to target learner abilities and even differentiate learning. These three coding options make it easy for teachers to use Sphero with students of all ages and abilities. Teachers do not need additional resources or activities to use these different canvases – the same Sphero used in a Year 1 classroom can also be used in a secondary school.



Why is Sphero Edu a Valuable Educational Tool?



STEAM Education

Sphero robots provide real life learning to science, technology, engineering, art, and maths. A few examples of the STEAM integrations built into the Sphero Edu learning activities are:

- Build a bridge and drive a Sphero robot across to test the structural integrity
- Create a long-exposure photograph
- Learn the maths and science behind Olympic sports to maximise a Sphero robot's long jump distance

In addition to the learning activities provided by Sphero, review the resources below for some ideas for integrating Sphero into STEAM education:

- [Teaching Physics with Sphero Robots](#)
- [How to Use Sphero the Robot for Incredible STEM Lessons](#)

Project Based Learning (PBL)

It is easy to expand your Sphero learning activities into PBL or include Sphero robots as a part of PBL. Are you new to PBL? Visit the Te Kete Ipurangi website to learn more and find a variety of resources including planning documents and rubrics:

- [Buck Institute for Learning](#)
- [Buck Institute for Education Essential Project Design Elements](#)

Personalised Learning

Content, method and pace can all vary in a personalised learning environment. The hands-on nature of Sphero makes it a perfect tool to tailor to the preferences, interests and pace of various learners.

Review these resources to learn more about personalised learning:

- [Personalised vs Differentiated vs Individualised Learning](#)
- [3 Ways to Personalise the Learning Experience](#)

Why is Sphero Edu a Valuable Educational Tool?

Computational Thinking

Sphero is the perfect platform to help students develop computational thinking skills and the mindsets that are necessary to compete in a global, technology-rich economy.

Sphero serves as both a coding platform and self-contained robotics system that can be used by any teacher or student, without any background in computer science. In addition, Sphero robots have a sophisticated set of sensors (called an inertial measurement unit or IMU) used for measuring forces and gathering data during scientific experiments.

Here are some examples of how Sphero activities help build a computational thinking mindset, with or without writing code.

Computational Thinking Fundamentals	What This Means	Examples in Sphero Activities
Decomposition	Does the activity encourage the student to break a larger problem into smaller problems to come up with a solution?	Students solve complex problems through smaller, more manageable tasks.
Pattern Recognition	Does the activity encourage the student to identify common patterns?	Students identify common patterns like movement, speed, light, time, or direction of the robot.
Pattern Generalisation and Abstraction	Does the activity encourage the student to make connection about common patterns?	Students connect concepts, such as speed & direction to how far the robot travelled.
Algorithm Design	Does the activity encourage the student to create logical steps that can be automated based on those patterns and connections?	Students create programs to control the Sphero robot. These often require using patterns like loops, which can be used to automate repeated behaviour.

Why is Sphero Edu a Valuable Educational Tool?

Explore the following computational thinking resources for more ideas:

- [ISTE's Computational Thinking Toolkit](#)
- [Hour of Code Activities](#) - Use Sphero robots to make hour of code activities come to life!

21st Century Skills

Sphero is the perfect platform to help students develop the mindsets that are necessary to compete in a global, technology-rich 21st century economy. Integrating Sphero into learning activities provides an opportunity to enhance 21st century skills such as creativity, collaboration, critical-thinking, and communication. The Sphero Edu App allows collaboration with other users around the globe to innovate the world of education and empower anyone to program. Refer to the [Framework for 21st Century Learning](#) for more information and definitions of 21st century skills.



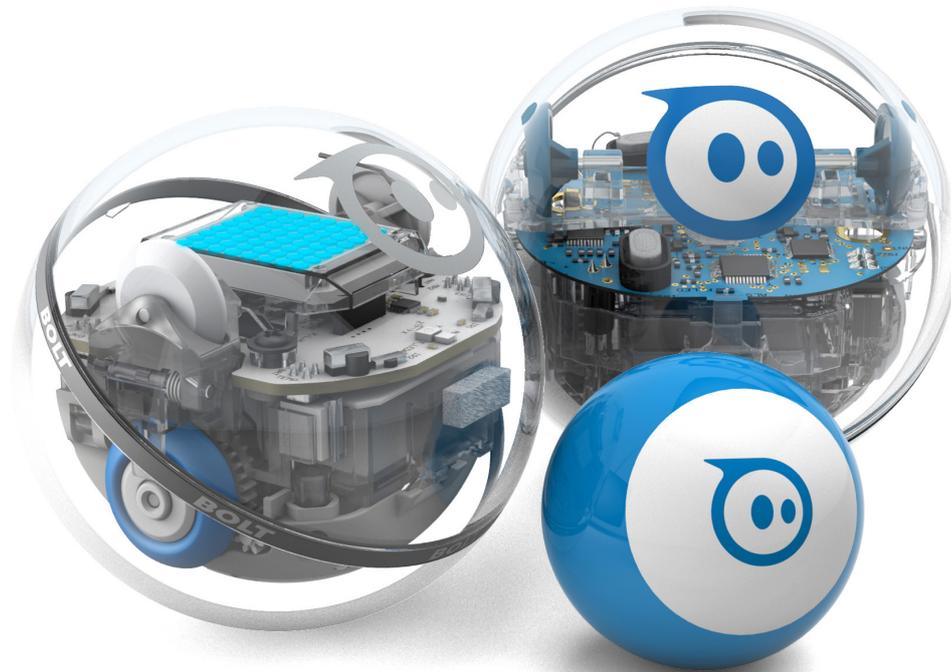
Why BOLT?



Australia's F-10 curriculum is designed to tap into children's and young people's natural inventiveness and their desire to create and work in practical ways. Sphero BOLT's features provide a new wave of opportunities for teachers and students to developing skills, knowledge, understanding and attitudes, and so maximise achievement.

- Well-designed activities which utilise the added features of Sphero BOLT offer children and young people opportunities to develop:
- Curiosity and problem-solving skills, a capacity to work with others and take initiative
- Planning and organisational skills in a range of contexts
- Creativity and innovation, for example through ICT and computer aided design and manufacturing approaches
- Skills in using tools, equipment, software and materials
- Skills in collaborating, leading and interacting with others
- Critical thinking through exploration and discovery within a range of learning contexts
- Discussion and debate
- Searching and retrieving information to inform thinking within diverse learning contexts
- Making connections between specialist skills developed within learning and skills for work
- Evaluating products, systems and services
- Presentation skills

Get to Know Your Robot



Inside the Robot

Sphero robots are approachable and simple to use, yet are packed with incredibly complex tech. Here's the gist of the magic inside your ball:

- **Circuit board** - The printed circuit board (or PCB) is what houses all of the electronics in your robot that process commands into actions. A Bluetooth chip within that board connects to your device, receives your commands, and sends them to the IMU, or the robot's brain, to process. Also built into the circuit board are the gyroscope and accelerometer, which detect your robot's movements, acceleration, and turning, helping to keep it oriented and driving where you tell it to drive. BOLT introduces an additional ambient light sensor that measures light intensity; a magnetometer (digital compass); infrared emitters and receivers for BOLT to BOLT communication, following, and evading; and a gorgeous 8x8 LED matrix that allows for custom images and animations.
- **Electric motor** - An electric motor turns the wheels that move your Sphero robot while the pressure from a stabiliser on top allow the wheels to move your robot, keeping it from going in circles inside the ball.
- **Charger** - To keep the Sphero robot's tech sealed inside its shell, it uses inductive charging rather than wiring to the batteries inside. Place it on the base and it'll charge right through the polycarbonate.

Get to Know Your Robot

Compatible Devices

Sphero robots must have an accompanying device to operate the robots. Here are some tips regarding these devices:

- A list of compatible devices for each robot type may be found at <https://www.sphero.com/devices>.
- [Chromebooks](#), [macOS](#), and [Windows 10](#) devices are now supported by Sphero Edu.
- The larger the screen, the better.
- If you are in need of mobile devices to run your Sphero robots, retired smartphones work great.
- Consider asking for old smartphone donations from your community.
- Remember to keep the mobile devices updated.
- Remember to keep the Sphero Edu app and other Sphero-related apps you may be using on the mobile devices updated.

Charging

Follow these tips to get the longest battery life out of your Sphero robot!

- How to charge your Sphero robots:
 - Sphero robots charge via Micro-USB cables and dedicated AC wall plugs. Computers can be used to charge Sphero robots as well but typically this will increase charging time due to the lower voltage output of a computer USB port.
 - Place the Sphero robot on the charging cradle heavy side down. Sphero robots will not charge if they are not placed heavy side down in the charger. Ensure students know how to place robots in the charger.
 - Plug power cord into a wall outlet. The blinking blue charger lights indicate the Sphero robot is charging.
 - ▶ If students are placing Sphero robots in the chargers, check at the end of class for the blue blinking light.
 - Charge for 3 hours or until the blue charger light stops blinking. It is OK to keep Sphero robots on the charger for longer. Remove robots from the cradle to get the party started.
- Make sure to charge your Sphero robots and mobile devices the night before use.
- Battery life slightly varies depending upon the specific robot. SPRK+ and Mini will give you around an hour of play, while BOLT is ready for over 2 hours of learning. Keep in mind battery life when planning Sphero robot use in your learning setting.
- It is best to fully drain Sphero robots by using until the battery is low, and then fully charging for about 3 hours or until the blue charging indicator light stops blinking. If you repeatedly charge and play with partial charges it will decrease the battery life.

Get to Know Your Robot

- Keeping Sphero robots on the charger for long periods of time will not decrease battery life.
- If you are going to store your robots for longer than a week, we recommend putting the robot to deep sleep by following the simple steps below:
 - ▷ Connect to the Sphero Edu app, tap on your robot name to see the options, then select "Turn Off."
 - ▷ Or, place on its plugged in charging base
 - ▷ Press and hold the button on the side of the charging base while simultaneously lifting the robot off of the base
 - ▷ While the robot is off the base, unplug the charger from the power source
 - ▷ Your robot is now in deep sleep!
 - ▷ To wake, simply place the robot on the charging base and plug it in
- Before connecting to the Sphero Edu app, ensure your robot is fully charged and not in deep sleep
- Do not store Sphero robots below 10 degrees C or above 26 degrees C (Doing so can lessen battery life).



Care and Maintenance

Here are some tips for storing your Sphero robots:

- Sphero robots are waterproof; to clean simply wipe your robot with warm soapy water and dry it with a towel.
- Sphero robots are also shockproof. Pop it, lock it, drop it. Your ball can handle it. That being said, we don't recommend testing this theory from the top of a tall building.

The Sphero Edu App

Sphero Edu is the Sphero app for programming Sphero robots. It is your hub to create, contribute, and go beyond code with Sphero. Sphero Edu makes it easy for educators, learners and parents to be involved. Learners may build programs and complete activities, educators manage classes and take learning beyond code, and parents may even create accounts for their kids.

What you can do with the Sphero Edu app:

- Drive your robot and change colours.
- Create programs for your robot using three canvas types: Draw, Block, and Text.
- Connect your robots to run programs you have created or search for a host of programs created by the Sphero team and community.
- Set up a classroom with student logins, assign activities and track them in real-time.
- Create an activity and assign it to your class, or share it publicly with the community.
- Find learning activities for different skill levels and content areas aligned to Common Core and Next Generation Science Standards.
- Collaborate with other users around the globe.
- Save your work by creating an account and jump between the apps or website.
- Find models of the Sphero robot to learn about the inner workings.
- Read the JavaScript Wiki to expand your coding knowledge!

Where to Get the App

The Sphero Edu app is available for free in the [Google Play](#) store and the [iTunes](#) store. The [Chrome App](#), [macOS app](#), and [Windows 10](#) are all available for download. After you've downloaded and installed the app, create an account so you can save your Programs and Activities in the cloud.

Connecting with Bluetooth

When you are ready to connect your robot, open the Sphero Edu app on a compatible mobile device and sign in to your account.

Connect your SPRK+, Mini, or BOLT to Sphero Edu by opening the Sphero Edu app, holding your robot next to your device, and selecting "Connect Robot." Find your robot model, and then select your specific robot in the list. Usually the robot you are holding is the first on the list.

If you have a Sphero 2.0 or Sphero SPRK, you need to double-tap it, then pair it in Bluetooth settings before opening the app.

The Sphero Edu App

How to Get Started

To get started, charge your Sphero robot. Once your robot is sufficiently charged, open the Sphero Edu app, and connect your robot using Bluetooth. Then, use the Drive functionality to make your Sphero robot roll forward.

App Features

The Sphero Edu app includes many features, including:

- **HOME**
 - ▷ Feed - Overview of user activity in the Sphero community
 - ▷ 3D Models - Learn the inner workings of your robot
- **ACTIVITIES**
 - ▷ All Activities - Find learning activities aligned to the Australian curriculum, AU-ACTD, when using the Desktop App, by filtering Edu Standards > Sphero > AU-ACTD.
 - ▷ My Activities - Learning activities created by you.
 - ▷ Workbook - Activities you have previously opened.
 - ▷ +Create - Create your own learning activities.
- **PROGRAMS**
 - ▷ All Programs - Pre-created programs for Sphero. Connect your robot and run!
 - ▷ My Programs - Programs you have created. Programs may be created in three different ways designed for learner progression:
 - ▶ **Draw:** Beginners can give robots commands by drawing a path that represents code for their robot to follow.
 - ▶ **Block:** Intermediate programmers can use code blocks to learn more advanced logic.
 - ▶ **Text:** Advanced programmers can use text programming and write their own JavaScript.
 - ▷ JavaScript Wiki - Expand your knowledge of JavaScript coding.
 - ▷ +Create - Create your own programs for Sphero robots.
- **CLASSES**
 - ▷ Learners - Add and manage your class list and class rosters.
 - ▷ Assignments - Assign activities to your classes. Review assignment progress.
 - ▷ Moderate - Find the status of activities and programs you have submitted to the Sphero Edu community.
- **DRIVE**
 - ▷ Connect, drive and change the colours of your robot.

Sphero Edu Activities



Introductory Activities

Are you wondering where to start with Sphero in your educational setting? The best way is to jump right in and learn by experimentation and play! The Sphero Edu team has developed a series of activities designed to introduce you and your learners to Sphero.

First, we recommend choosing a canvas to learn. The app interface in which you will control the Sphero robot is called a canvas– and to support a wide range of skills and abilities, we’ve developed three canvases: Draw, Block, and Text. All three canvases are available in the Sphero Edu app.

Not sure which one to start with? Here is more information about each canvas:

- **Draw** - Uses a drawing interface. Effective in Primary years 1-6 and all class types.
- **Block** - Uses a drag-and-drop block interface and teaches the logical structure of code. Effective in Primary years 4-6, Secondary years 7-12 and all class types.
- **Text** - Uses the programming language JavaScript. Effective in Secondary years 7-12 and classes that focus on computer science and programming.

The Sphero Edu team has developed a series of activities designed to introduce you and your learners to Sphero.

Getting Started with Draw

[Draw 1: Shapes](#): This lesson introduces learners to Sphero by challenging them to draw shapes that represent code and execute that code through the Draw canvas.

Getting Started with Blocks

[Blocks 1: Intro and Loops](#): This lesson introduces learners to Sphero through an overview of the app, how to create programs using block coding, and how to use loops and operators.

Getting Started with Text

[Text 1: Hello World](#): This lesson introduces learners to Sphero through an overview of the text canvas, how to use loops and operators, and tips for getting started with their first lines of JavaScript code.

Sphero Edu Activities

Classroom Management

To create a class in the Sphero Edu app, click on **Classes > Learners** and follow the directions to create a class and add your learners.

- To speed up the process, choose the "Add from Roster" option and upload a CSV file with the names of your learners.
- If you are a Google or a Clever user, you can automatically sync your classes. View more information here: <https://edu.sphero.com/about>.

If you need assistance visit the Sphero support website: <https://support.sphero.com/support/home>.

Sphero Edu Activities

If you would like to create your own activities in the Sphero Edu app, click on **Activities > Create**. Once created, you can find your activities under **Activities > My Activities**. To assign an activity to your class, navigate to the activity and click **Assign**.

Sphero Edu has a collection of learning activities found at <https://edu.sphero.com/cwists/category>. All Sphero Edu Activities follow this framework:

- **Exploration:** Activate learner prior knowledge related to the challenge. Consider starting Sphero activities "unplugged," meaning learners begin the activity by planning and brainstorming without the Sphero robots.
- **Skill-building:** Follow with learners completing a guided activity with Sphero to learn the skills needed for the challenge. Make sure to build in time for students to play, learn, and discover.
- **Challenge:** Learners use their new knowledge and skills to solve a problem utilising Sphero robots.

Whether creating your own activities or using the Sphero Edu learning activities, consider having additional supplies available for your Sphero learning activities. Example supplies could include cardboard, tape, scissors, paint, and other crafting supplies.

In addition, rubrics are a valuable tool for assessing learning when using Sphero robots. A Creativity and Innovation Rubric supplements Sphero well. Below are links to Creativity and Innovations Rubrics for different school levels from the [Buck Institute for Education](#):

- [K-2 Creativity and Innovation Rubric](#)
- [3-5 Creativity and Innovation Rubric](#)
- [6-12 Creativity and Innovation Rubric](#)

Are you an Educator in Australia? Good news, you can now search learning activities aligned to the Australian curriculum, AU-ACTD, when using the Desktop App, by filtering Edu Standards > Sphero > AU-ACTD

Sphero Edu Activities

Primary School

The activities below require knowledge of the Draw or Blocks canvas. Many of these lessons have a slight focus on different curriculum areas as indicated. You can find more lessons designed for years 1-6 on the Sphero Edu app or website.

- Draw 1 - 3 (Introductory Activities)
- Blocks 1 - 4 (Introductory Activities)
- What a Character (Language and Storytelling)
- The Heart (Science)
- Area of Rectangles (Maths)
- Maze Mayhem (General)
- Perimeter (Maths)
- Light Painting (Art)

Upper Primary/Lower Secondary School

Most of the activities below require knowledge of the Blocks canvas. Many of these lessons have a slight focus on different curriculum areas as indicated. You can find more lessons designed for Primary year 6 and Secondary years 7-9 on the Sphero Edu app or website.

- Blocks 1 - 4 (Introductory Activities)
- What a Character (Language and Storytelling)
- The Heart (Science)
- Secret Message (Social Studies)
- Planetary Motion (Science)
- Helmets for the Win! (Science)
- Avoid the Minotaur (General)

Secondary School

Most of the activities below require knowledge of the Blocks or Text canvases. Many of these lessons have a slight focus on different curriculum areas as indicated. You can find more lessons designed for Secondary years 7-10 on the Sphero Edu app or website.

- Blocks 1 - 4 (Introductory Activities)
- Text 1 - 4 (Introductory Activities)
- The Heart - High School (Science)
- Atom Tracks - High School (Science)
- Fortune Teller - High School (Maths)
- Morse Code - Data Structures (Computer Science)

Sphero Edu Activities

Engineering and Robotics Activities

Are you teaching engineering or robotics? The activities below encourage teamwork, collaboration, and creativity. Many of them employ open challenges that require students to practice engineering design principles and make use of room, time, or material constraints.

Activities to explore:

- Avoid the Minotaur
- Bridge Challenge
- Chariot Challenge
- Maze Mayhem
- Sphero City
- Tractor Pull
- Swim Meet
- Jousting Tournament
- Build a Sphero Run

Coding Sequence

Are you teaching your learners to code? Follow the sequence below to introduce your learners to computer science fundamentals, JavaScript syntax, and industry standards like pseudocoding, debugging, and refactoring.

Note: These activities are considered advanced and are primarily geared for students in grades 9-12. While you do not have to complete these in the order below, many of them build upon the skills gained in previous activities.

Activities to explore:

1. Text 1: Hello World
2. Text 2 - Conditionals
3. Text 3 - Loops & Variables
4. Text 4 - Functions & TDD
5. Morse Code - Data Structures
6. Fun, Fun, Functions
7. Recursion & Ocean Colourss

Units of Learning - Ancient Egypt



P3 - 5: Ancient Egypt - Computational Thinking with Sphero
Minimum of three charged Sphero robots, compatible devices, and the Sphero Edu app needed.

By the end of the unit pupils will be able to:

- 1) design, write and debug programs that accomplish specific goals; solve problems by decomposing them into smaller parts;
- 2) use logical reasoning to explain how simple algorithms work and to detect and correct errors in algorithms and programs;
- 3) use sequence, selection, and repetition in programs;
- 4) use technology safely, respectfully and responsibly

	Lesson 1	Lesson 2	Lesson 3	Lesson 4
Sphero Robot	SPRK+, enhanced with BOLT	SPRK+, enhanced with BOLT	SPRK+, enhanced with BOLT	SPRK+, enhanced with BOLT
Australian Curriculum	<p>Follow, describe and represent a sequence of steps and decisions (algorithms) needed to solve simple problems. ACTDIP004</p> <p>Identify and explore a range of digital systems with peripheral devices for different purposes, and transmit different types of data ACTDIK007</p> <p>Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them. ACTDIP010</p>	<p>Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input. ACTDIP011</p> <p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p> <p>Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language. ACTDIP030</p>	<p>Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them. ACTDIP010</p> <p>Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input. ACTDIP011</p> <p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p>	<p>Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them. ACTDIP010</p> <p>Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input. ACTDIP011</p> <p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p>
Outcomes	<ul style="list-style-type: none"> I can use the draw tools to draw a triangle that meets at three points. 	<ul style="list-style-type: none"> I can practice refactoring my program. I can create and execute a Blocks program. 	<ul style="list-style-type: none"> I can create and execute a Blocks program. 	<ul style="list-style-type: none"> I can use the light instructions to create and execute a Blocks program.
Subject Focus	Maths	Maths	Social Sciences	Music
Curriculum Links	Shape	Position & Movement/Angles	People in History	Listening and Timing
Activity Outline and Steps	<p>Draw with Sphero An activity to introduce new users to the draw tools in SpheroEDU.</p> <p>Can student draw a triangle and get a Sphero robot to follow the trail?</p> <p>Can students change colours for each of the three movements?</p> <p>Can students see the relationship between using the draw tool and then experimenting with the block editor to create the same image?</p>	<p>Coding the river Nile Use Sphero tape to create several mini- golf like courses around the room.</p> <p>Pupils work in pairs to code their programmes to complete each challenge.</p>	<p>Hieroglyph Writing Students create hieroglyph symbols using a Sphero robot.</p> <p>Using various angles and colours as well as calculated measurements of speed and time. Students create code which allows a Sphero robot to outline different symbols. Encourage students to draw out their symbol first and label the turns and angles.</p> <p>Use the "Slow Shutter" app to photograph the hieroglyph symbols.</p>	<p>Dance Like an Egyptian Watch 'Uptown Funk' demonstration.</p> <p>Ask Students - How do you think they got the lights to dance to the music?</p> <p>Listen to the song several times (Walk like an Egyptian by The Bangles) and select a section you want to use, and share this section with students.</p> <p>Students use timed light instructions to create a disco lightshow for this song.</p>
Challenge	Can you re-create the same shape using coding instructions?	How few attempts can you use to complete each course accurately?		
Resources	Large open space	Tape Various maze challenges Themed image cut-outs (Download A)	Hieroglyph guide (Download B) Slow Shutter App	
Assessment	Observations & Programmes – Are children able to take this step?	Observations & Programmes	Compare 'Slow Shutter' photographs – can they be deciphered?	Class disco – switch off lights and start all Sphero robots at once, to the music of course!

Units of Learning - Ancient Egypt



P3 - 5: Ancient Egypt - Computational Thinking with Sphero

Minimum of three charged Sphero robots, compatible devices, and the Sphero Edu app needed.

By the end of the unit pupils will be able to:

- 1) design, write and debug programs that accomplish specific goals; solve problems by decomposing them into smaller parts;
- 2) use logical reasoning to explain how simple algorithms work and to detect and correct errors in algorithms and programs;
- 3) use sequence, selection, and repetition in programs;
- 4) use technology safely, respectfully and responsibly

	Lesson 5	Lesson 6	Lesson 7	Lesson 8
Sphero Robot	SPRK+, enhanced with BOLT	SPRK+, enhanced with BOLT	SPRK+, enhanced with BOLT	BOLT
Australian Curriculum	<p>Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them. ACTDIP010</p> <p>Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input. ACTDIP011</p> <p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p>	<p>Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them. ACTDIP010</p> <p>Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input. ACTDIP011</p> <p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p>	<p>Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them. ACTDIP010</p> <p>Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input. ACTDIP011</p> <p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p> <p>Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors. ACTDIP029</p>	<p>Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input. ACTDIP011</p> <p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p> <p>Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language. ACTDIP030</p>
Outcomes	<ul style="list-style-type: none"> I can create and execute a Blocks program. I can convey information using speech instructions. 	<ul style="list-style-type: none"> I can create and execute a Blocks program. I can convey information using speech instructions. 	<ul style="list-style-type: none"> I can develop and communicate imaginative design solutions. 	<ul style="list-style-type: none"> I can practice refactoring code. I can define and use loops. I can create and execute a Blocks program.
Subject Focus	Social Sciences	Social Sciences	Social Sciences	Social Sciences
Curriculum Links	People in History	People in History	Instructional Writing	People in History
Activity Outline and Steps	<p>Rolling Tour: Egyptian Gods Using the Sphero Edu app's ability to speak given text, Students will create code which will instruct a Sphero robot to move from one Egyptian God to another, providing a fact about each.</p> <p>Students will use the provided worksheets at check-points to stop and discuss each of the Egyptian Gods.</p> <p>If using Sphero BOLT, students could use the 8x8 LED matrix program scrolling text of each Egyptian God's name, or create unique animations for each.</p>	<p>Mummification Students use the mummification cards to show their understanding of mummification.</p> <p>Students lay cards out on the floor and programme a Sphero robot to travel to the cards in order.</p> <p>The 8x8 LED matrix could be used to indicate the numbered steps: first, second, third etc, or to create a symbol for each step.</p>	<p>Senet Game Challenge Watch tutorial video: How to play Senet https://www.youtube.com/watch?v=ikfRHqW46jE</p> <p>Students recreate the game using Sphero robots rather than stones.</p> <p>Students have to create their own game board.</p>	<p>Hieroglyph Maze Using their understanding of block coding, students enable a Sphero robot to navigate a maze created by the teacher using tape on the floor. The Hieroglyph cards indicate instructions that should be programmed into the LED matrix within the created code.</p>
Challenge			Can you write a set of instructions – how to play your game?	
Resources	Egyptian God cards (Download C)	Mummification Cards (Download D)	https://www.youtube.com/watch?v=ikfRHqW46jE	Hieroglyph Cards (Download E)
Assessment	Invite other classes to come and enjoy the tour – does it work?	Assess understanding of process as well as code creation.	Are visitors able to play the game using their instructions?	Observations & Programmes

Units of Learning - Space



P4 - 7: Space - Computational Thinking with Sphero

Minimum of three charged Sphero robots, compatible devices, and the Sphero Edu app needed.

By the end of the unit pupils will be able to:

- 1) design, write and debug programs that accomplish specific goals; solve problems by decomposing them into smaller parts;
- 2) use logical reasoning to explain how simple algorithms work and to detect and correct errors in algorithms and programs;
- 3) use sequence, selection, and repetition in programs;
- 4) use technology safely, respectfully and responsibly

	Lesson 1	Lesson 2	Lesson 3	Lesson 4
Sphero Robot	SPRK+, enhanced with BOLT	SPRK+, enhanced with BOLT	SPRK+, enhanced with BOLT	SPRK+, enhanced with BOLT
Australian Curriculum	<p>Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them. ACTDIP010</p> <p>Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input. ACTDIP011</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p>	<p>Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them. ACTDIP010</p> <p>Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input. ACTDIP011</p> <p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p>	<p>Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them. ACTDIP010</p> <p>Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input. ACTDIP011</p> <p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p>	<p>Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them. ACTDIP010</p> <p>Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input. ACTDIP011</p> <p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p>
Outcomes	<ul style="list-style-type: none"> I can solve a problem by decomposition. I can plan an enquiry. 	<ul style="list-style-type: none"> I can design a programme to make a circle. I can describe the movement of the Earth, Moon & Sun. 	<ul style="list-style-type: none"> I can write and debug a programme. I can listen with attention to detail and recall sounds. 	<ul style="list-style-type: none"> I can use logic and key vocab to explain a programme. I can describe positions on the full coordinate grid.
Subject Focus	Science	Science	Music	Maths
Curriculum Links	Planet Earth	Planet Earth	Dynamics of Music	Shape, Position, & Movement
Activity Outline and Steps	<p>Rocket launch Students explore movement and controls. What can a Sphero robot do and not do?</p> <p>Watch rocket launch video.</p> <p>Decompose task: launch a Sphero robot like a rocket in small groups (speaks countdown, rolls quickly & stops).</p> <p>Use movement, controls and sound to programme.</p> <p>Film demos.</p>	<p>Our solar system Recap Sun, Moon & Earth orbits using students as planets.</p> <p>Demo of Shape Shifter program and explore editing variables.</p> <p>Discuss how to make circle (spin & speed block).</p> <p>Students programme at least three Sphero robots to model their two orbits.</p> <p>Design programmes in pairs and then test in small groups.</p> <p>Review each other's programmes.</p>	<p>Star Wars Show Students design a light show to music.</p> <p>Watch an example from the Sphero Edu app that uses Uptown Funk.</p> <p>Use the Star Wars opening credits music, in 15 second chunks.</p> <p>Design light show in rhythm to music. Share your programme with other groups.</p>	<p>Discover Mars Students explore Mars' surface using Google Earth.</p> <p>On large grid, design a route for the rover from the Habitation Zone to find water.</p> <p>Students programme a chosen route to discover water.</p> <p>Narrate journey and add comments to blocks.</p>
Challenge	Can you add a "blast off" sound to your programme?	Can you make the Sun, Moon & Earth different colours that fade?	Can you explain why it's easier to use the strobe block over multiple LED blocks?	Can you reflect the location of one rover on the grid in another quadrant?
Resources	Magic 8 demo Blocks launch template	Shape Shifter demo Blocks circle template	Uptown Funk demo Music & sound	Google Mars demo Large 'Mars' grids with Habitation Zone & hidden water/ice
Assessment	Invite other classes to come and enjoy the tour – does it work?	Opportunity to assess understanding of process as well as code creation.	Observations - Are visitors able to play the game using their instructions?	Observations & Programmes

Units of Learning - Space



P4 - 7: Space - Computational Thinking with Sphero

Minimum of three charged Sphero robots, compatible devices, and the Sphero Edu app needed.

By the end of the unit pupils will be able to:

- 1) design, write and debug programs that accomplish specific goals; solve problems by decomposing them into smaller parts;
- 2) use logical reasoning to explain how simple algorithms work and to detect and correct errors in algorithms and programs;
- 3) use sequence, selection, and repetition in programs;
- 4) use technology safely, respectfully and responsibly

	Lesson 5	Lesson 6	Lesson 7	Lesson 8
Sphero Robot	SPRK+, enhanced with BOLT			
Australian Curriculum	<p>Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them. ACTDIP010</p> <p>Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input. ACTDIP011</p> <p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p>	<p>Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them. ACTDIP010</p> <p>Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input. ACTDIP011</p> <p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p>	<p>Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them. ACTDIP010</p> <p>Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input. ACTDIP011</p> <p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p>	<p>Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them. ACTDIP010</p> <p>Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input. ACTDIP011</p> <p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p>
Outcomes	<ul style="list-style-type: none"> I can use the if/then blocks to include conditionals. I can explain day and night referring to Earth's rotation. 	<ul style="list-style-type: none"> I can use functions to create a programme. I can explore the patterns and sounds of language. 	<ul style="list-style-type: none"> I can test out and edit a programme for a specific goal. I can explore the patterns and sounds of language. 	<ul style="list-style-type: none"> I can detect and correct errors to debug a programme I can apply attacking and defending principles.
Subject Focus	Science	MFL & Literacy	MFL & Literacy	Physical Education
Curriculum Links	Planet Earth	Modern Languages	Modern Languages	Team Games
Activity Outline and Steps	<p>Earth Quiz Research Earth facts safely & recap Earth's rotation related to day/night.</p> <p>Play Simon Says to explore If/then commands.</p> <p>Explore If/Then and If/Else blocks for Simon Says.</p> <p>Students choose 5 Earth questions and design a programme for a quiz. What happens when some answers right/wrong?</p>	<p>Alien language 1 Watch 'Arrival' clip and discuss symbol language.</p> <p>Students will choose 5 common words and design a symbol to represent each word.</p> <p>Use functions to create a programme for the new language.</p> <p>Students will paint their language symbols by covering a Sphero robot in paint and running their programme.</p>	<p>Alien language 2 Students will explore other languages that use symbols for whole words. Look how certain words are written in Kanji. Try to decode the words man, tree, river.</p> <p>Students will run their programmes made in part 1 for another group to decode.</p> <p>As an added bonus, use a long-exposure app to capture each sybmol's light path.</p>	<p>Meteor crash Watch meteor shower video, and discuss how students would defend the Earth from such an event.</p> <p>Explore If/Else conditionals and the event called "on collision."</p> <p>Review the Earth's orbit from lesson 2 and design a programme for Meteors.</p> <p>Take time to debug their programme before trying to defend or attack the Earth with their programmes.</p>
Challenge	Can you programme a Sphero robot to change colour if you throw it?	Can you make a sentence using your new language?	Can you use glow in the dark paint to create a different effect?	Can you programme a Sphero robot to swim backstroke? Take it swimming!
Resources	Blocks template Earth facts Rotation demo	Function templates Pens, paints & large paper Symbol examples	Kanji examples Part 1 programmes Long exposure app	Blocks template to debug
Assessment	If/then exploration Programmes Review of others'	Symbols designed Functions Painted symbols	Guessing each other's programmes Long exposure images	Programmes Detecting errors Attack/defence

Units of Learning - World



P4 - 7: World - Computational Thinking with Sphero

Minimum of three charged Sphero robots, compatible devices, and the Sphero Edu app needed.

By the end of the unit pupils will be able to:

- 1) design, use & evaluate computational abstractions that model the state & behaviour of real-world problems & physical systems;
- 2) understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem;
- 3) understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming.

	Lesson 1	Lesson 2	Lesson 3	Lesson 4
Sphero Robot	SPRK+, enhanced with BOLT	SPRK+, enhanced with BOLT	SPRK+, enhanced with BOLT	SPRK+, enhanced with BOLT
Australian Curriculum	<p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p> <p>Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language ACTDIP030</p>	<p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p> <p>Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors. ACTDIP029</p> <p>Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language. ACTDIP030</p> <p>Define and decompose real-world problems precisely, taking into account functional and non-functional requirements and including interviewing stakeholders to identify needs. ACTDIP038</p>	<p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p> <p>Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors. ACTDIP029</p> <p>Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language. ACTDIP030</p>	<p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p> <p>Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors. ACTDIP029</p> <p>Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language. ACTDIP030</p>
Outcomes	<ul style="list-style-type: none"> I can describe the interdependence of organisms in a food web. 	<ul style="list-style-type: none"> I can research a different culture to understand user needs. 	<ul style="list-style-type: none"> I can describe and imitate the work of Lee Krasner. 	<ul style="list-style-type: none"> I can describe the causes of the Opium Wars and the journey from Britain.
Subject Focus	Science	Social Studies	Art	Social Studies & Health
Curriculum Links	Biology & Ecosystems	Design and Engineering	Expressive Art	World History & Drug Abuse
Activity Outline and Steps	<p>Ocean food webs Discover Sphero and what it can and can't do.</p> <p>Students explore food webs to better understand producers and consumers.</p> <p>Introduce the concept of sequencing & its importance when creating algorithms.</p> <p>Draw large ocean web & use a Sphero robot to show energy transfer.</p>	<p>Ugandan water Watch the video of people walking for water in Uganda.</p> <p>As a class, discuss sanitation issues around the world and brainstorm solutions. Explore pulleys and how to build a well.</p> <p>Create pulley that can lift a Sphero robot from the ground to table height.</p>	<p>American art Look through the work of Lee Krasner. Have students share their thoughts and reactions to her art.</p> <p>Explore different programmatic algorithms and predict which will produce certain shapes.</p> <p>Test each of the different algorithms and compare their results with your.</p> <p>Students will add multiple algorithms to a loop to recreate some of Krasner's paintings.</p>	<p>Chinese opium Watch the mini-documentary on the 1830s Opium wars, or discuss key points with students.</p> <p>How does this historical issue compare to current issues surrounding recreational drug use and the inherent dangers.</p> <p>Students will role play as traders from the UK travelling, trading, and warring with the Chinese.</p> <p>Using a "smuggling map," show the traders' journey.</p> <p>Study the If/Else programmes that show the traders' journey carrying tea/opium. Debug your chosen algorithm and narrate Sphero's journey.</p>



P4 - 7: World - Computational Thinking with Sphero

Minimum of three charged Sphero robots, compatible devices, and the Sphero Edu app needed.

By the end of the unit pupils will be able to:

- 1) design, use & evaluate computational abstractions that model the state & behaviour of real-world problems & physical systems;
- 2) understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem;
- 3) understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming.

	Lesson 1	Lesson 2	Lesson 3	Lesson 4
Challenge	Can you model bee pollination using similar algorithms?	Can you work out how much water is equivalent to a Sphero robot's weight?	Can you make a Sphero robot imitate the art of Yayoi Kusama?	Can you programme one Chinese and one British Sphero robot to combat?
Resources	Videos A2 paper & pens	Video Building materials (e.g. kebab sticks, bottle stops, string)	Algorithm examples (Paints & paper, use Location data images)	Template algorithms Large map of the world traced on A1
Assessment	Discussion of food webs Algorithms	Research Group discussions Pulley models	Predictions Loops Evaluation	Role play Discussion of programmes Algorithm and narration

Units of Learning - World



P4 - 7: World - Computational Thinking with Sphero

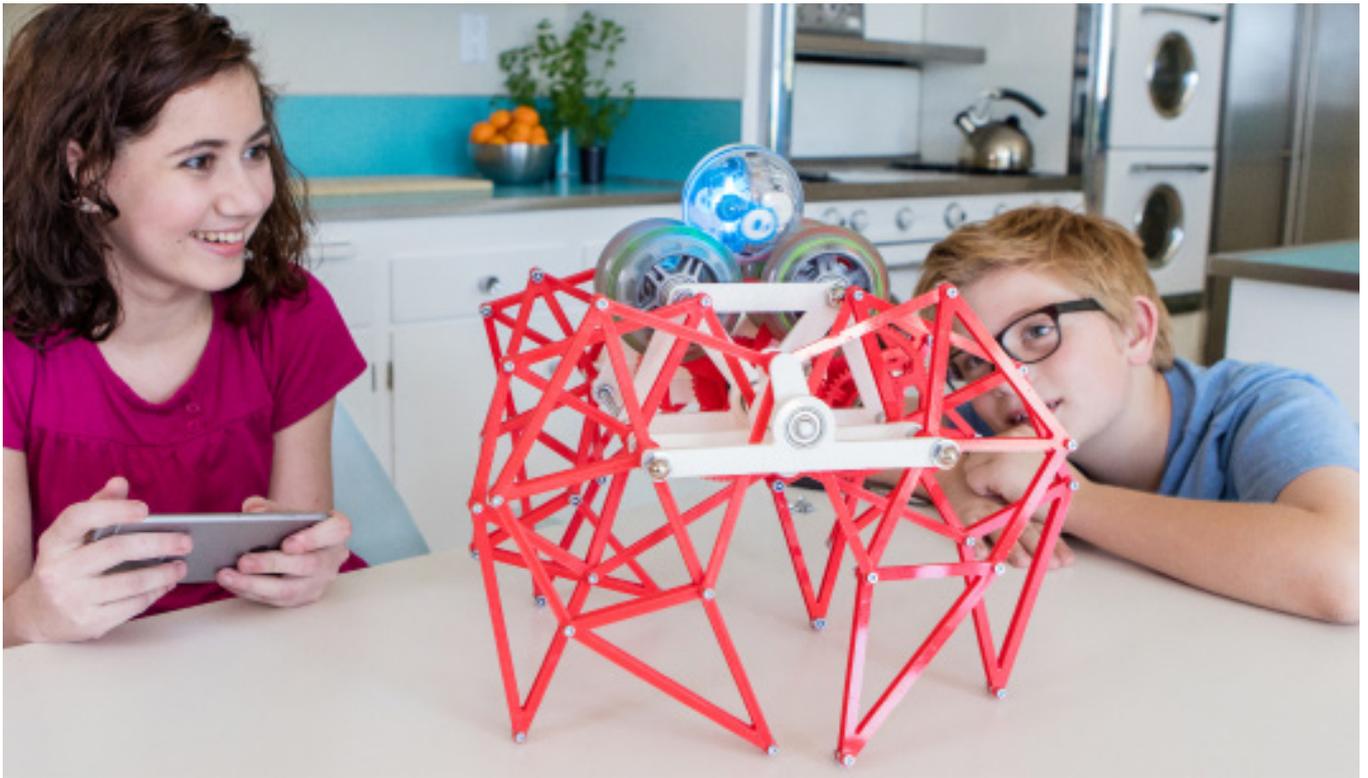
Minimum of three charged Sphero robots, compatible devices, and the Sphero Edu app needed.

By the end of the unit pupils will be able to:

- 1) design, use & evaluate computational abstractions that model the state & behaviour of real-world problems & physical systems;
- 2) understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem;
- 3) understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming.

	Lesson 5	Lesson 6	Lesson 7	Lesson 8
Sphero Robot	SPRK+, enhanced with BOLT			
Australian Curriculum	<p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p> <p>Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors. ACTDIP029</p> <p>Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language. ACTDIP030</p>	<p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p> <p>Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors. ACTDIP029</p> <p>Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language. ACTDIP030</p>	<p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p> <p>Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors. ACTDIP029</p> <p>Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language. ACTDIP030</p>	<p>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). ACTDIP019</p> <p>Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input. ACTDIP020</p> <p>Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors. ACTDIP029</p> <p>Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language. ACTDIP030</p>
Outcomes	<ul style="list-style-type: none"> I can use loops in my models. I can describe the results of translation, rotation and reflection. 	<ul style="list-style-type: none"> I can design programmes that model problems. I can explain how human processes influence environments. 	<ul style="list-style-type: none"> I can explain why and how Boolean logic is used. 	<ul style="list-style-type: none"> I can use Boolean logic.
Subject Focus	Maths	Geography	Science	Geography
Curriculum Links	Shapes	Human / Physical Geography	Biology	Fieldwork
Activity Outline and Steps	<p>British painters Explore the work of Bridget Riley. Identify examples of mathematical translation, rotation and reflection.</p> <p>Students will discuss use of loops in other programs and how they could be used to recreate Riley's art work.</p> <p>Devise a way to attach a marker to a Sphero robot. Create programmes that create patterns similar to her art. Run the programmes with the marker attached. What was the result?</p>	<p>Costa Rican turtles Review If/Else statements.</p> <p>As a class, watch the video of turtle egg poaching in Costa Rica. Discuss the consequences to human interference and other human influence on the environment.</p> <p>Discuss what an accelerometer is and how it can be used in their programmes.</p> <p>Programme a Sphero robot to fool the poachers by rolling to the sea (use If/Else statements to overcome holes and hills in sand.)</p>	<p>Indian mammals Watch the endangered animals video and identify the animals native to India.</p> <p>Introduce boolean logic through the 20 questions programme.</p> <p>Have students predict what happens when use they AND/OR/NOT in a search engine. Use a desired search engine to search for different Indian animals using boolean logic.</p> <p>Compare each other's search results and discuss their observations.</p>	<p>Global code What are the differences between a globe and an atlas? Spend some time making observations of the African continent.</p> <p>Review boolean logic together as a class. Pick an African country and describe it using Boolean logic (e.g. It is north of Sudan AND east of Libya.)</p> <p>Students will use the programme template that speaks AND/OR/NOT at random to complete sentences. Use this to describe the country you chose and try to guess your classmates'.</p>
Challenge	Can you work out how to see your code written in JavaScript?	Can you make a Sphero robot jump around to scare the poachers?	Can you programme a Sphero robot to jump like a frog?	Can you make a Sphero robot travel across a globe visiting different countries?
Resources	Markers, plastic cups, and tape	Videos Sand pit (ideally!) with holes and hills	Videos Internet enabled devices for web browser searches	Template programme Globes & atlases
Assessment	Discussion of geometric transformations Programmes	Programmes Discussion of human factors	Discussion on extinction Search results	Descriptions Guessing others'

Supplemental Resources



Sphero Created Activities

All Sphero Edu activities can be found at <https://edu.sphero.com/cwists/category>. Find an activity that meets your learning objectives and continue learning with Sphero!

Thursday Learn Day

On Thursdays, the Sphero Edu team posts new programs that you can try with your learners! You can find these programs here: <https://edu.sphero.com/remixes>.

Troubleshooting Website

You can find support for most Sphero issues on our support website: <https://support.sphero.com/support/home>.

If you are having connection troubles, try the following strategies:

- If the robot does not connect to Sphero Edu, place your robot on the charger for 15 seconds to ensure it's not in deep sleep, then try again.
- If your robot is disconnecting often and you are in a room with a lot of users, try turning off wifi and bluetooth on the devices that are not being used with a robot. Limiting a room to about 20 robots and programming devices or less is a good rule of thumb.

Supplemental Resources

Javascript Wiki

The Sphero Edu documentation is located here: <https://sphero.docsapp.io/docs/get-started>. This wiki is a guide for students and teachers to learn how to program Sphero robots with JavaScript, the most common web programming language in the world.

Research / Case Studies

Programming with Sphero Edu's app-enabled robots fosters skills that lead to a wide variety of careers. However, when it comes to mainstream coding instruction, demand still outweighs supply. We're here to close the gap and help bring the future into classrooms today.

STEM and STEAM curriculum are a critical part of Primary and Secondary education, and superlative approaches to sharpening 21st-century skills will hinge on better coding instruction. The best solutions include coding that is interactive and fun, but also strategies that go beyond code by incorporating robotics and technology with collaborative STEAM activities, nurturing students' imaginations in new and exciting ways. These types of solutions appear to be moving the needle for local education authorities looking to improve coding instruction. According to the CDE survey, respondents who use Sphero solutions were more likely to say that coding instruction was meeting students' needs.

Instead of looking at coding as a requirement to fulfill, Primary and Secondary schools should consider it an open-ended form of art. Better coding instruction sparks divergent thinking and creativity; these concepts, coupled with the real-world benefits of learning how to program, will change the playing field for every student in Australia over the next five years and beyond. Read more about coding and STEM/STEAM skills in our schools by reading [Cracking the Code: Six keys to better coding instruction in K-12 education](#), a case study with Sphero in the K-12 classrooms in the US.

Marketing Resources

The brand assets for Sphero Edu are located at brandfolder.com/spheroedu. This Brandfolder is for educators and community members seeking logos, images and general up to date information about Sphero Edu.

Administrator Guide

Share the [Administrator Guide](#) with principals, curriculum directors, instructional technologist and any other administrators at your school that want to learn more about Sphero Edu.

Supplemental Resources

Makerspace Guide

Review the [Makerspace Guide](#) for specific ideas for integrating Sphero robots into maker learning.

Social Media

Connect with Sphero Edu online for ideas, tips, and resources:

- Facebook: <https://www.facebook.com/GoSphero/>
- Twitter: <https://twitter.com/spheroedu>
- Instagram: <https://www.instagram.com/sphero/>
- Pinterest: <https://www.pinterest.com/sphero/>

Sphero Blog

Visit our education blog for updates, tips, and suggestions: <https://medium.com/@SPRK>.

Support

- [Support Home](#)
- [Contact Us](#)

Security & Privacy

We are dedicated to ensuring Sphero Edu is safe and secure to use. Some of our 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>.