Sphero - Young Inventors

Year level band: 7-8

Description: In this lesson students will explore the use of Sphero in the everyday world by adding accessories to invent solutions to workplace or other problems or simply by inventing an adaptation to the device. In each case, they are to build the accessory and create the code required for the device to serve a particular purpose.

Type: This lesson is a transition from visual programming to general purpose programming. It explores decomposition, branching, iteration and functions in SPRK.

Type: Design, Visual Programming functions

Resources:

- Sphero robot
- Tickle software app for iOS and Android devices
- Sphero Curriculum available online
- Introductory video of the Sphero and Meet Sphero
- Building and construction materials such as Knex or Lego, cardboard, bottle tops, tape, plastic cups, scissors, tape, straws etc
- Sphero Apps including
    - SPRK Lightning Lab - Programming for Sphero Robots by Orbotix Inc. (Sprk+)
    - Sphero by Sphero Inc. (Sprk+)
    - Sphero Exile by Sphero, Inc.
    - Sphero ColorGrab by Sphero Inc.
    - Sphero Edu by Sphero Inc.
- Examples of adaptations to Sphero - eg Chariot races, Sphero examples on Pinterest

Prior Student Learning:

Students have been exposed to Sphero using play-based learning, and are able to create simple algorithms using Lightning Lab.

Digital Technologies Summary

Students design adaptations to the Sphero Ball involving adding other materials to create a new product. Students develop skills in designing a solution for a user with specific needs, and being able to communicate their design intentions with sketch designs, as well as verbally by sharing their designs with peers.

Students will be encouraged to follow the Design and Technologies process:

- investigating and defining
- generating and designing
- producing and implementing
- evaluating
- collaborating and managing.

By reflecting on their own designs, as well as other designs, students develop skills in being able to evaluate designs and provide constructive feedback.

They direct their own learning, plan and carry out investigations, and become independent learners who can apply **design thinking**, technologies understanding and skills when making decisions.

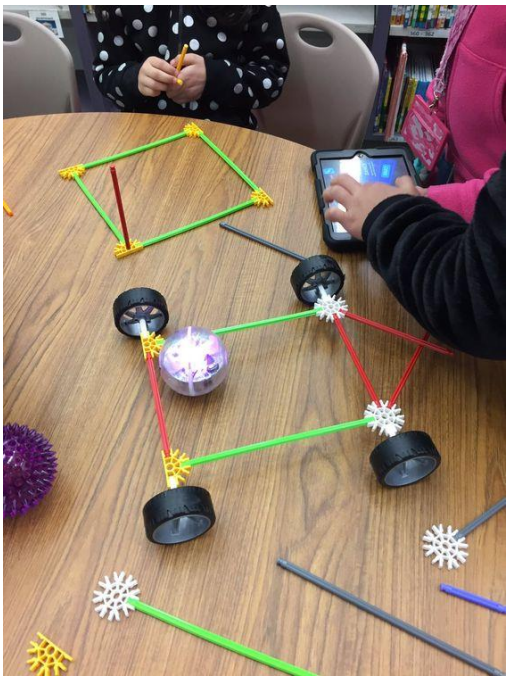
Designing and innovation involve a degree of risk-taking and as students work with the uncertainty of sharing new ideas they develop resilience. (Personal and Social Capability)

Band	Content Descriptors
Year 7-8	**Digital Technologies** Define and decompose real-world problems taking into account functional requirements and economic, environmental, social, technical and usability constraints (ACTDIP027) • determining the factors that influence proposed solution ideas, for example identifying limitations on the weight that the Sphero can pull Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language (ACTDIP039) • identify and clarify how functions can be developed using SPRK, and how these functions can be used with parameters • design, implement, run, and modify a SPRK program that achieves the planned outcome • decompose an algorithm into key functions, and write and test these functions on the Sphero • modify the program to suit optional challenges **Science** People use science understanding and skills in their occupations and these have influenced the development of practices in areas of human activity (ACSHE121)
	Critical and Creative Thinking Inquiring - identifying, exploring and organising information and ideas • Pose questions: pose questions to probe assumptions and investigate complex issues Generating ideas, possibilities and actions • Imagine possibilities and connect ideas: draw parallels between known and new ideas to create new ways of achieving goals • Consider alternatives: generative alternatives and innovative solutions, and adapt ideas, including when information is limited or conflicting • Seek solutions and put ideas into action: predict possibilities and identify and test consequences when seeking solutions and putting ideas into action Reflecting on thinking and processes • Think about thinking (metacognition): assess assumptions in their thinking and invite alternative opinions • Reflect on processes: evaluate and justify the reasons behind choosing a particular problem-solving strategy Analysing, synthesising and evaluating reasoning and procedures • Evaluate procedures and outcomes: explain intentions and justify ideas, methods and courses of action, and account for expected and unexpected outcomes against criteria they have identified
Achievement standards	• Students evaluate information systems and their solutions in terms of meeting needs, innovation and sustainability • Students define and decompose problems in terms of functional requirements and constraints • Students design user experiences and algorithms incorporating branching and iterations, and test, modify and implement digital solutions.

Element	Summary of tasks
Learning hook	How can Sphero be used to solve problems the real world? How can Sphero be adapted to perform a specific task? What is the algorithm for the program that will run on the Sphero? Ask the students to think about tasks that are repetitive or complex, or requiring precision for example, a medicine case needs to be moved across a map, going to specific points to deliver the medicine. Introduce the students to the concept of functions, showing how functions dramatically reduce the number of lines of code we have to write. For each algorithm, students work in groups to identify the basic building blocks for achieving the task: • How many building blocks can students identify? • What are the advantages/disadvantages of using each of them? Introduce the concept of function parameters and how these are used within functions.

Learning Map
 	
(Sequence)
	1. Revisit previous knowledge of Sphero
	2. Introduce and attempt challenge
	3. Students write the pseudo code for the algorithm that will achieve the planned outcome
	4. Students identify the key building blocks for the algorithm
	5. For example, for a Sphero that transports weight between different points on a map, the key building block could be:
	  • roll in a straight line for a while, corresponding to x meters (x is a parameter)
	  • make a turn if a variable is set
	6. The algorithm would then see the repetition of this building block a number of times, with different parameter values.
	7. This building block is called a function, say *move*
	8. Students write the algorithm using repeated calls to *move*:
	  • move(5)
	  • move(4)
	  • ...
	9. Students write the SPRK program and test the program in groups
	10. Students their solutions to the brief in group discussion format

Learning input	• Revisit knowledge of Sphero technology and coding • Discuss possible device adaptations which could ‘add value’ to its usage • Discuss the shortcomings of this device, eg. it’s just a ball • Allow time for discussion on Bluetooth and comparisons to Wireless. Discuss wireless, Bluetooth and connected networks. • Introduce or elaborate on visual programming languages and the importance of clear instructions (Algorithms) when controlling Sphero and setting tasks. • Discuss benefits of using functions. • Discuss challenges of using functions.
Learning construction	Students are now asked to consider an adaptation to the Sphero that will serve a new purpose eg carrying a passenger, leaving a track (eg texta), pushing material, pulling material A variety of materials will be provided to provoke thoughts and ideas. Students must respond to the following brief: • Adapt Sphero by adding accessories to invent solutions to workplace or other problem.

	Define what your machine is designed to do, and create a code that allows Sphero to perform the required task. Eg. Sphero pulls 500g of weight 1m to serve the purpose of an automated wheelbarrow. Write the pseudocode for your algorithm, identifying building blocks that could be used as functions. Students can do this by first decomposing the behaviour into main blocks, and then identifying behaviour that could be suitable to be implemented as a function: • Is the Sphero doing something repeatedly? • Is the Sphero doing something where some key parameters change? Write the code for the Sphero and load the code on it. Students test and debug the code in pairs. Think about answers to the following questions when you present your design to the class: • What problem is being solved? • Why is this design useful? • What did you hope your design would be able to achieve? Did your code work to achieve this? • What worked and what didn't work in your design? • What could have made your design better? The rest of the class are required to ask informed questions or provide constructive feedback to the presenting group to demonstrate students' ability to reflect upon their work and respond to feedback.
Learning demo	Demonstrate examples of Sphero adaptations (photos, videos) such as the ones below. Be aware that showing examples may lead to 'copying' with thought. Copying with variations will be acceptable.   
Learning reflection	**Think about answers to the following questions when you present your design to the class:** • **What problem is being solved?** • **Why is this design useful?** • **What did you hope your design would be able to achieve? Did you code work to achieve this?** • **What worked and what didn't work in your design?**

	• **What could have made your design better?** The rest of the class are required to ask informed questions or provide constructive feedback to the presenting group to demonstrate students' ability to reflect upon their work and respond to feedback.

Assessment:

Formative Assessment

- Teachers could collect evidence of learning and progression, eg. plans, evidence of the design process
- Student demonstrations explaining their product; ability to reflect upon their work and respond to constructive feedback
- Ability to ask informed questions or provide constructive feedback to peers

	Quantity of knowledge			**Quality of understanding**	
Criteria	**Pre-structural**	**Uni-structural**	**Multi-structural**	**Relational**	**Extended abstract**
Algorithms Programming	No visual program written within app interface.	Algorithm only shows a limited number of instructions but do not allow Sphero to progress or connect.	Algorithm has enough instructions to complete the task but not linked to Sphero or is not relevant to the defined goal of the machine.	Algorithm has instructions linked in the correct sequence to achieve the task – Sphero can follow a path as planned and designed in related to the defined goal of the machine.	Algorithm brings in prior learning and/or independent learning beyond the task and possibly includes additional blocks and features (e.g. loops), code matches the defined goal of the machine. Full use of Programming interface is evident
Design	No design used	Basic design with no features identified, does not solve any identifiable problem	Basic design with some features identified, partly linked to design justification	Detailed design with numerous features identifies and linked to design justification	Detailed design that brings in prior learning and/or independent learning beyond the task and possibly includes requirements, specifications, constraint factors. Clearly solves an identified problem and well linked to design justification.
Vocabulary	No specific / technical terms used.	The terms program or code may be used as a general description.	The terms program or code are used as a general description. The terms analogue and digital are known and used correctly.	Specific terms such as program, loop, debug are used confidently with specific reference to learner's work. Code is commented in specific places.	Understanding of specific terms such as constant, function, parameter, and variable.

Teacher/Student Instructions:

The Sphero robot is a versatile device in many ways although does not have sensors as in some robots for example Meet Edison or Dash and Dot. However there is much that can be done with the Sphero within a visual programming framework using the Tickle app and or MacroLab. Please note that the Sphero Sprk+ is not compatible with all Apps

Bluetooth connectivity can be tricky at times, particularly with a number of other bluetooth devices in the same vicinity. Be sure to leave space enough for students to connect to the right Sphero.

CSER Professional Learning:

This lesson plan corresponds to professional learning in the following CSER Digital Technologies MOOCs:

Year 7-8 Next Steps:

- Unit 3 Data & Visualisation: Problem Definition and Design
- Unit 2 Next Steps 7 & 8

See: http://csermoocs.adelaide.edu.au/moocs

Further Resources:

There are many Sphero apps now available to explore with your students on the iPad, more being added over time. Here are some of the ones you might explore with your students. Only a few are currently compatible with Sphero SPRK+

- SPRK Lightning Lab - Programming for Sphero Robots by Orbotix Inc. (Sprk+)
- Sphero by Sphero Inc. (Sprk+)
- Sphero Exile by Sphero, Inc.
- Sphero ColorGrab by Sphero Inc.
- Sphero Edu by Sphero Inc.

Digital Technologies Hub: www.digitaltechnologieshub.edu.au

CSER: https://csermoocs.adelaide.edu.au/moocs/

Author: Celia Coffa