

# Sphero - Invent a game

**Year level band:** 7/8

**Description:** In this lesson students will explore the use of Sphero by creating a game. In each case, they are to build an accessory and design and build a game board/space and create a game that requires players to code to play. The audience and aim of the game is to be explicit alongside any relevant rules and instructions.

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

## Resources:

- Sphero or Sphero Sprk+ robot
- 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, pipe cleaners etc (general construction materials)
- Sphero Apps including
  - Sphero Macrolab by Orbotix Inc.
  - SPRK Lightning Lab Programming for Sphero Robots by Orbotix Inc.
- Examples of adaptations to Sphero eg Chariot races, Sprk 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. They are familiar with the coding aspects to enable basic control of a sphero eg - move in a square.

## Digital Technologies Summary

Students design a game based on using a Sphero and create a new product and games space. 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	<p data-bbox="475 152 735 181"><b>Digital Technologies</b></p> <p data-bbox="475 221 1353 282">Design the user experience of a digital system, generating, evaluating and communicating alternative designs (ACTDIP028)</p> <ul data-bbox="504 311 1485 611" style="list-style-type: none"> <li data-bbox="504 311 1485 387">● Identify features that make a HID interface easy to use and incorporate these into their own design</li> <li data-bbox="504 423 1485 499">● Identifying similar digital systems and their user interfaces, assessing whether user interface elements can be reused.</li> <li data-bbox="504 535 1485 611">● Presenting and comparing alternative designs to a solution for a problem, for example presenting alternative design mockups to the class</li> </ul> <p data-bbox="475 647 1501 707">Implement and modify programs with user interfaces involving <u>branching</u>, <u>iteration</u> and functions in a general-purpose programming language (ACTDIP030)</p> <ul data-bbox="504 743 1497 1066" style="list-style-type: none"> <li data-bbox="504 743 1497 819">● identify and clarify how functions can be developed using SPRK, and how these functions can be used with parameters</li> <li data-bbox="504 846 1497 922">● design, implement, run, and modify a SPRK program that drives a Sphero through a maze using functions</li> <li data-bbox="504 949 1497 1025">● decompose an algorithm into key functions, and write and test these functions on the Sphero</li> <li data-bbox="504 1039 1497 1066">● modify the program to suit optional challenges</li> </ul> <p data-bbox="453 1099 831 1128"><b>Critical and Creative Thinking</b></p> <p data-bbox="453 1155 1270 1184">The particular elements of Critical and Creative Thinking addressed:</p> <p data-bbox="453 1216 959 1245">Generating ideas, possibilities and actions</p> <ul data-bbox="504 1279 1474 1503" style="list-style-type: none"> <li data-bbox="504 1279 1474 1355">● Imagine possibilities and connect ideas: draw parallels between known and new ideas to create new ways of achieving goals</li> <li data-bbox="504 1359 1474 1435">● Consider alternatives: generative alternatives and innovative solutions, and adapt ideas, including when information is limited or conflicting</li> <li data-bbox="504 1440 1474 1503">● Seek solutions and put ideas into action: predict possibilities and identify and test consequences when seeking solutions and putting ideas into action</li> </ul> <p data-bbox="453 1536 903 1565">Reflecting on thinking and processes:</p> <ul data-bbox="504 1599 1430 1664" style="list-style-type: none"> <li data-bbox="504 1599 1430 1664">● Transfer knowledge into new contexts: justify reasons for decisions when transferring information to similar and different contexts</li> </ul> <p data-bbox="453 1697 1238 1727">Analysing, synthesising and evaluating reasoning and procedures</p> <ul data-bbox="504 1760 1474 1825" style="list-style-type: none"> <li data-bbox="504 1760 1474 1825">● Apply logic and reasoning: identify gaps in reasoning and missing elements in information</li> </ul>

Element	Summary of tasks
Learning hook	<p>What makes a good game?</p> <p>Present some examples of good games that could include the Sphero. For example, in PacMan, the Sphero can play the role of monsters.</p> <p>What elements are essential? Optional?</p> <p>How can we use Sphero to create a game that requires programming to play (not simply using the Drive tool)?</p> <p>Ask the students to think about tasks that are repetitive or complex in the game: for example, if the game is a ponglike game, the Sphero would need to move for a while in one direction, then move into another, etc this type of behaviour is a good candidate for using functions.</p> <p>Introduce the students to the concept of functions, showing how functions dramatically reduce the number of lines of code we have to write.</p> <p>For each algorithm, students work in groups to identify the basic building blocks for achieving the task:</p> <ul style="list-style-type: none"> <li>• How many building blocks can students identify?</li> <li>• What are the advantages/disadvantages of using each of them?</li> </ul> <p>Introduce the concept of function parameters and how these are used within functions.</p>
Achievement Standards	<ul style="list-style-type: none"> <li>• Students define and decompose problems in terms of functional requirements and constraints</li> <li>• Students design user experiences and algorithms incorporating functions, branching and iterations, and test, modify and implement digital solutions.</li> </ul>
Learning Map (Sequence)	<ul style="list-style-type: none"> <li>• Revisit previous knowledge of Sphero</li> <li>• Introduce and attempt challenge of creating a game: <ul style="list-style-type: none"> <li>○ Rules</li> <li>○ Game Play</li> <li>○ Winning condition</li> <li>○ Integration of Sphero in the game play</li> </ul> </li> <li>• Team teach game to another group, self and peerassess</li> <li>• Challenge: students program the Sphero following the game rules <ul style="list-style-type: none"> <li>○ Students write the pseudocode for your algorithm, identifying building blocks that could be used as functions</li> </ul> </li> <li>• Is the Sphero doing something repeatedly?</li> <li>• Is the Sphero doing something where some key parameters change?</li> <li>• Students identify the key building blocks for the algorithm</li> <li>• For example, if a Sphero is playing the role of a monster moving in a maze, like in the game PacMan, the key building block could be: <ul style="list-style-type: none"> <li>○ roll in a straight line for a while, corresponding to x meters (x is a parameter)</li> <li>○ make a turn if a variable is set</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>• The algorithm would then see the repetition of this building block a number of times, with different parameter values.</li> <li>• This building block is called a function, say move</li> <li>• Students write the algorithm using repeated calls to move: <ul style="list-style-type: none"> <li>○ <i>move(5)</i></li> <li>○ <i>move(4)</i></li> <li>○ ...</li> </ul> </li> <li>• Students write the SPRK program and test the program in groups</li> <li>• Students write the code for the Sphero and load the code on it.</li> <li>• Students test and debug the code in pairs.</li> </ul>
Learning input	<ul style="list-style-type: none"> <li>• Revisit the coding and use of Sphero What is possible?</li> <li>• Discuss elements of a good game What is the aim of the game? Rules, instructions, variations</li> <li>• Discuss why/how Sphero may be a good addition to a game</li> </ul>
Learning construction	<p>Students are now asked to design a game using Sphero that matches the elements of a 'good game' discussed earlier. The choice of game is up to the students, however must be playable by a small groups (24 players). A variety of materials will be provided to provoke thoughts and ideas. Students work in small groups.</p>
Learning demo	<p>Demonstrate examples of Sphero games (photos, videos) such as the ones below. Be aware that showing examples may lead to 'copying' without thought. Copying with variations will be acceptable.</p> <div style="display: flex; justify-content: space-around;">   </div>
Learning reflection	<p>Groups must then combine and teamteach their game to another group. Students must selfassess and peer assess their own game and someone elses' based on the following criteria:</p> <ul style="list-style-type: none"> <li>• Ease to understand</li> <li>• Fun factor</li> <li>• Use of Sphero</li> <li>• Creativity</li> </ul> <p>Class discussion:</p> <ul style="list-style-type: none"> <li>• Was the game a mix of challenge and fun?</li> <li>• What sort of features can students think of to enhance the game?</li> </ul>

	<ul style="list-style-type: none"> <li>• What was difficult about creating this game?</li> <li>• What are the challenges about coding this game for the Sphero</li> <li>• What are the benefits of using functions in this case?</li> <li>• Was this game easy to test/debug?</li> <li>• What were the challenges when designing the functions?</li> <li>• What were some of the advantages?</li> <li>• What other things do you need to consider when writing a program with functions?</li> <li>• Was it difficult to program the Sphero using functions?</li> <li>• What are the advantages and disadvantages of using functions?</li> <li>• What was challenging about using functions?</li> </ul>
--	--

**Assessment:**

Formative Assessment

- Teachers could collect evidence of learning and progression, eg. plans, evidence of the design process
- Ability to teach game to another group and respond to questions from the group
- Self- and peer-assessment

Criteria	Quantity of knowledge			Quality of understanding	
	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 well linked to Sphero	Algorithm has instructions linked in the correct sequence to achieve the task – Sphero can follow a path/behaviour in the game as designed	Algorithm brings in prior learning and/or independent learning beyond the task and possibly includes additional blocks and features, or new functions Full use of Programming interface is evident

Design	No game design used	Basic game design with no features identified, Sphero not well utilised	Basic/good game design with some features identified, Sphero adequately utilised	Detailed game design with numerous features identifies, Sphero well utilised	Detailed game design that brings in prior and/or independent learning beyond the task and possibly includes requirements, specifications, constraint factors. Sphero utilised extremely well throughout.
Presentation	Poor description and demonstration of game	Fair description of game missing clear instructions	Reasonably good description of game with only a few instructions missing	Good, clear description of the game, including demonstrations, with no missing instructions	Excellent description, clear instructions and useful demonstrations, explains every required element of the game with little ambiguity.
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 Sphero 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:

CSER F-6 Digital Technologies: Extended

- Unit 1 - Fundamental thinking skills

Year 7-8 Next Steps

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

See: <http://csermoots.adelaide.edu.au/moots>

## Further Resources:

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

Here are some of the ones you might explore with your student

[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: <http://www.digitaltechnologieshub.edu.au>

CSER: <https://csermoocs.adelaide.edu.au>

