Sphero Maze

Year level band: Year 7-8

Description:

This lesson will explore how to program the Sphero using functions and show the benefits of decomposing the behaviour of the Sphero into functions, instead of writing line by line repeated behaviours.

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

Resources:

- Sphero robot
- Tickle software app for iOS and Android devices
- Sphero Curriculum available online
- Introductory video of the Sphero and Meet Sphero
- Masking tape to mark out robot paths.
- Building and construction materials such as Knex or Lego
- 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.

Prior Student Learning:

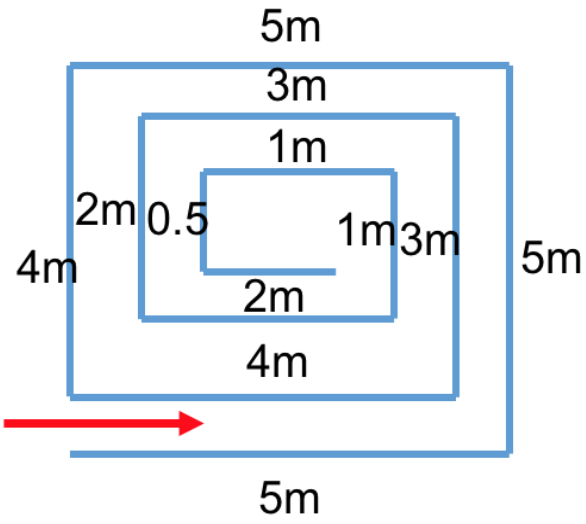
Students have been exposed to Sphero using play-based learning, and are able to create simple algorithms using SPRK Lightning Lab. They are familiar with the coding aspects to enable basic control of a Sphero eg move in a square.

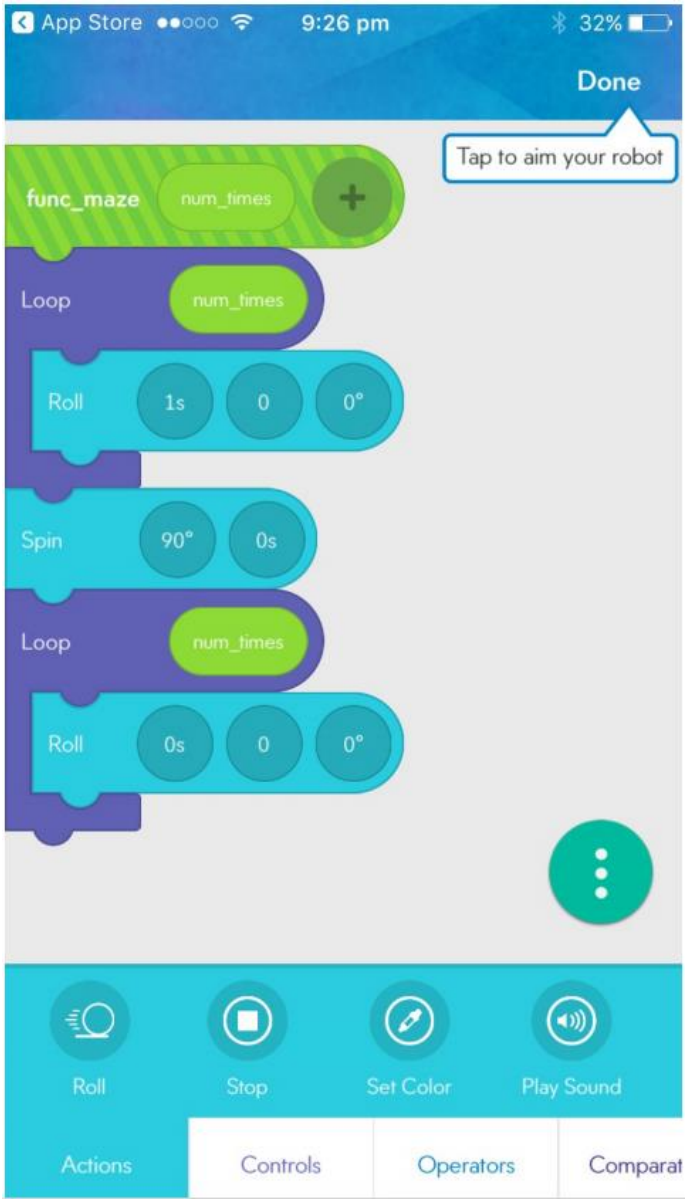
Digital Technologies Summary

Students broaden their programming experiences to incorporate subprograms into their solutions. They predict and evaluate their developed and existing solutions, considering time, tasks, data and the safe and sustainable use of information systems, and anticipate any risks associated with the use or adoption of such systems.

Band	Content Descriptors
Year 7 and 8	• Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language (ACTDIP030)

	○ 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 drives a Sphero through a maze using functions ○ decompose an algorithm into key functions, and write and test these functions on the Sphero ○ modify the program to suit optional challenges
	Critical and Creative Thinking Inquiring – identifying, exploring and organising information and ideas • Identify and clarify information and ideas • Organise and process information Generating ideas, possibilities and actions • Consider alternatives • Seek solutions and put ideas into action Analysing, synthesising and evaluating reasoning and procedures • Apply logic and reasoning
Achievement Standards	By the end of Year 8, students plan and manage digital projects to create interactive information. They 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	Discuss with students mazes: • Have you ever seen a garden maze? • What would be the best way to get out of/in a circular maze? • What about other mazes? Ask students to devise an algorithm to get the Sphero to the center of this maze:  Ask them how many instructions they would have to use to get the Sphero inside the center of this maze. • What if the maze was bigger, say, double in size? Introduce the students to the concept of functions, showing how functions dramatically reduce the number of lines of code we have to write. Identify the basic building blocks for getting to the centre of this maze: • 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)
	• Students build their SPRK program without using functions
	• Students build their SPRK program using functions, identifying the parameters, and necessary building blocks
	• The key building block could be:
	  ○ roll in a straight line for a while, corresponding to x meters (x is a parameter)
	  ○ turn left
	  ○ roll in a straight line for a while, corresponding to x meters (x is a parameter)
	• The algorithm would then see the repetition of this building block a number of times, with different parameter values.
	• This building block is called a function, say func_maze
	• Students write the algorithm using repeated calls to func_maze:
	  ○ func_maze(5)
	  ○ func_maze(4)
	  ○ ...
	
	 

Learning input	The teacher introduces the Sphero or Spheros to the class. Have the students discuss how Sphero can be sent instructions. 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 will work collaboratively to plan out their algorithm and write the program with and without functions. Students will then work together to test and debug their programs. Students can test their programs for different maze sizes. As an extra challenge, students could devise mazes for other teams, and test and debug programs. Another challenge can be asking students to write their program with the smallest number of blocks in the main part.
Learning demo	Once the lesson is complete, students will be able to move their Sphero to the center of the maze and show the repeated use of functions.
Learning reflection	Encourage students to reflect on the process: • What were the challenges when designing the function? • What were some of the advantages? • What other things do you need to consider when writing a program with functions? • What did you learn from creating this? • What were the fun moments? • Was it difficult to program the Sphero? • What are the advantages and disadvantages of using functions? • What was challenging about using functions?

Assessment:

Formative Assessment

- Teachers observe students using the Spheros, creating their algorithms and debugging.
- Use questioning to elicit student understanding of algorithms and their algorithmic thinking.
- You might take photos/videos of the students' work to document their progress.

	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. Algorithm has enough instructions to complete tasks but functions are not used	Algorithm has instructions linked in the correct sequence to achieve the task – Sphero can follow a path as designed, and functions are used.	Algorithm brings in prior learning and/or independent learning beyond the task and possibly includes additional blocks and features (e.g. loops, functions). Full use of Programming interface is evident.
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 Sadh 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.

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:

- F-6 Digital Technologies: Foundations
    - Unit 7: Algorithms and Programming
- F-6 Digital Technologies: Extended
    - Unit 2: Algorithms & Programming Extended
- 7-8 Next Steps
    - Unit 2 - Next Steps
    - Unit 3 – Problem definition and design
    - Unit 4 – Implementation and assessment

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.

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: Claudia Szabo