**Sphero: Catch me if you can**

*Please refer to the online lesson plan on the DT Hub to access all website links and additional resources.*

**Year level 5-6:** In this lesson students will use Sphero to explore Bluetooth connectivity, algorithms through maze mapping and visual programming language.

**Description:**​ ​**Resources:**

* Sphero robot
* Sphero Curriculum available online
* Introductory video of the Sphero and Meet Sphero
* Building and construction materials such as Knex or Lego
* Sphero Apps including
* Sphero: Getting started
* Sphero Bolt + lessons
* Sphero Edu apps
* Hour of code

**Prior Student Learning:**

By years 5 and 6 many students may have had some experience with a visual programming language such as Scratch or Blockly (which is the basis of the Hour of Code).

If your students are new to visual programming languages there are a number of excellent resources they might try to help them become familiar with computational thinking tasks. The Hour of Code is a good place to start.

|  |  |
| --- | --- |
| **Digital Technologies Summary** | |
| By the end of Year 6, students will have had opportunities to create a range of digital solutions, such as games or quizzes and interactive stories and animations. For example controlling a robotic device to follow a set of instructions.  Students increase the sophistication of their algorithms by identifying repetition and incorporate repeat instructions or structures when implementing their solutions through visual programming, such as reading user input until an answer is guessed correctly in a quiz. They evaluate their solutions and examine the sustainability of their own and existing information systems.  The use of robots such as Sphero provide students with an opportunity to work collaboratively with their peers to solve problems such as following a maze through the programming of step by step instructions. | |  |  |  |
| **Band** | **Content Descriptors** |
| **5 - 6** | Design algorithms involving multiple alternatives (branching) and iteration (AC9TDI6P02).  Implement algorithms as visual programs involving control structures, variables and input (AC9TDI6P05). |
| The particular elements of Critical and Creative Thinking addressed by this content description:   * 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 6, students will have had opportunities to create a range of digital solutions, such as games or quizzes and interactive stories and animations. |

|  |  |
| --- | --- |
| **Element** | **Summary of tasks** |
| Learning hook | Sphero is wanting to find his/her friend who has become stuck in a tight spot in a corner of the classroom and cannot find a path out through the maze of objects scattered on the path.  What set of instructions could you give Sphero to help navigate to his/her friend and then help them both to return?  Alternatively, students may wish to explore a particular seafarer or land explore from the past by drawing a large stylised map on the school playground and have Sphero take on the role of Captain Cook of Burke and Wills for example.  What adventures might he/she have on his/her voyage? This could work in nicely with a history or geography lesson on significant historical figures navigating parts of a geographical landscape. |
| Learning Map  (Sequence) | * Students work together to plan a solution to a problem * Students create a series of instructions (Algorithms) with alternative solutions should one plan fail. * Students connect to robot using bluetooth device to send instructions. * Students test visual program algorithm and debug until successful. * Students create a game that could be shared with their peers. Instructions could be written for the game that others could use to learn how to play the game. |
| Learning input | The teacher introduces the Sphero or Spheros to the class and asks them what they think it might be.  After a number of responses and the eventual answer being a robot that can be controlled by a digital device, 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 instructructions (Algorithms) when controlling Sphero and setting tasks. |
| Learning construction | Explore how to create instructions using the Sphero Edu apps. Move the Sphero along a path. Send instructions for the Sphero to follow.  Students, once they are comfortable with the visual programming language, might set up paths using masking tape or objects for Sphero to negotiate. Have students work in teams to solve a number of challenges. For example, in this lesson students follow a square pathway that is in the shape of a square.  To introduce the lesson, the teacher briefly revise the concept of a ‘regular’ shape and the language used to describe polygons Greek prefixes for numbers.  Students worked in teams of about five to follow step-by-step instructions to ‘draw’ Square and Triangle, then Extension to create obscure paths using learned skills.  Students can create programs in which the Sphero changes colour and speed.  Create a maze, or obstacle course and then program Sphero to navigate the course (see Maze Mayhem ). Opportunity for team learning in the areas of measurement, problem solving, and programing using Sphero Edu app .  Plan and implementing a solution using a visual programming language, for example designing and creating a simple computer game involving decisions and repetitions, suitable for younger children, that requires user input to make selections, taking into account user responses. |
| Learning demo | Students create their own program to solve a challenge. Provide challenges for students to choose from or as a starting point to design their own challenge.   * Shape challenge: Program the Sphero to navigate a more complex shape (triangle, pentagon, or star). In this challenge students may use iteration (loops). * Obstacle course navigation: Set up small obstacles (cups, cones, books) and have students program the Sphero to avoid them while reaching a target. Introduce ‘checkpoints’ that they must pass through. Include a gamification element if they pass through a section then they score 1 point. * Speed and distance race: Create a racetrack and program the Sphero to the optimal speed to race the course within a set time. C * Colour changes: Program the Sphero to change colours at different points in the journey, for example, passing through a tunnel, turning corners, reaching the finish line. |
| Learning reflection | Students could look at the limitations and advantages of a spherical robot. Could such a robot be used for supporting rescue missions or exploring unfamiliar environments?  What sort of features can students think of to enhance Sphero’s capabilities?  Are students able to explain in plain English their instructions (algorithms) to others and carry out testing of algorithm to check for bugs. Can they then debug their algorithm if problems are found?  Were the students engaged and prepared to solve the problems they came up against when creating an algorithm for Sphero to complete the challenge presented? |

**Assessment:**

Formative Assessment

* Observe students' initial engagement with Sphero
* Review their ability to determine how Sphero can receive instructions via Bluetooth
* Observe engagement and understanding of visual programming language using Tickle or MacroLab apps.
* Demonstrate understanding of use of visual programming language within and app to control a wireless or Bluetooth device.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Quantity of knowledge** | | | **Quality of understanding** | |
| **Criteria** | **Pre-structural** | **Uni-struct ural** | **Multi-struct ural** | **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 instructions  linked in the correct sequence to achieve the task​ – Sphero can follow a path as designed | Algorithm brings in prior learning and/or independent learning beyond the task and possibly includes additional blocks and features (e.g. loops)  Full use of  Programming  interface is evident |
| Vocabulary | When  describing algorithm, no specific vocabulary is used | The terms instruction or code may be used as a general description | The term algorithm is used as a general description | The terms algorithm and program is used confidently with specific reference to learner’s work | Specific vocabulary like decisions and repetition is used, going beyond the  set language |

**Further Resources:**

* Download entire Lesson Plan
* See Overview Section for Sphero resources
* Computer Science Education Research Group (CSER)
* This lesson plan corresponds to professional learning in the following CSER Digital Technologies MOOCs :
* F-6 Digital Technologies: Foundations

Unit 7: Algorithms and Programming

Unit 8: Visual Programming

* F-6 Digital Technologies: Extended

Unit 2: Algorithms & Programming Extended