**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 play­based 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 risk­taking and as students work with the uncertainty of sharing new ideas they  develop resilience. (Personal and Social Capability) | 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 risk­taking and as students work with the uncertainty of sharing new ideas they develop resilience. (Personal and Social Capability) |

|  |  |
| --- | --- |
| **Band** | **Content Descriptors** |
| **Year 7­8** | **Digital Technologies**  Design the user experience of a digital system [(AC9TDI8P07)](https://v9.australiancurriculum.edu.au/f-10-curriculum.html/learning-areas/digital-technologies/year-7_year-8/content-description?subject-identifier=TECTDIY78&content-description-code=AC9TDI8P07&detailed-content-descriptions=0&hide-ccp=0&hide-gc=0&side-by-side=1&strands-start-index=0&subjects-start-index=0&view=quick).  Generate, modify, communicate and evaluate alternative designs [(AC9TDI8P08)](https://v9.australiancurriculum.edu.au/f-10-curriculum.html/learning-areas/digital-technologies/year-7_year-8/content-description?subject-identifier=TECTDIY78&content-description-code=AC9TDI8P08&detailed-content-descriptions=0&hide-ccp=0&hide-gc=0&side-by-side=1&strands-start-index=0&subjects-start-index=0&view=quick).  Design algorithms involving nested control structures and represent them using flowcharts and pseudocode [(AC9TDI8P05)](https://v9.australiancurriculum.edu.au/f-10-curriculum.html/learning-areas/digital-technologies/year-7_year-8/content-description?subject-identifier=TECTDIY78&content-description-code=AC9TDI8P05&detailed-content-descriptions=0&hide-ccp=0&hide-gc=0&side-by-side=1&strands-start-index=0&subjects-start-index=0&view=quick). |
| **Critical and Creative Thinking**  The particular elements of Critical and Creative Thinking addressed:  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:   * Transfer knowledge into new contexts: justify reasons for decisions when transferring information to similar and different contexts   Analysing, synthesising and evaluating reasoning and procedures   * Apply logic and reasoning: identify gaps in reasoning and missing elements in information |

|  |  |
| --- | --- |
| **Element** | **Summary of tasks** |
| Learning hook | What makes a good game?  Present some examples of good games that could include the Sphero. For example, in PacMan, the Sphero can play the role of monsters.  What elements are essential? Optional?  How can we use Sphero to create a game that requires programming to play (not simply using the Drive tool)?  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.  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. |
| Achievement Standards | * Students define and decompose problems in terms of functional requirements and constraints * Students design user experiences and algorithms incorporating functions, branching and iterations, and test, modify and implement digital solutions. |
| Learning Map (Sequence) | * Revisit previous knowledge of Sphero * Introduce and attempt challenge of creating a game:   + Rules   + Game Play   + Winning condition   + Integration of Sphero in the game play * Team teach game to another group, self­ and peer­assess * Challenge: students program the Sphero following the game rules   + Students write the pseudocode for your algorithm, identifying building blocks that could be used as functions * Is the Sphero doing something repeatedly? * Is the Sphero doing something where some key parameters change? * Students identify the key building blocks for the algorithm * 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:   + roll in a straight line for a while, corresponding to x meters (x is a parameter)   + make a turn if a variable is set * 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 move * Students write the algorithm using repeated calls to move:   + *move(5)*   + *move(4)*   + *...* * Students write the SPRK program and test the program in groups * Students write the code for the Sphero and load the code on it. * Students test and debug the code in pairs. |
| Learning input | * Revisit the coding and use of Sphero What is possible? * Discuss elements of a good game What is the aim of the game? Rules, instructions, variations * Discuss why/how Sphero may be a good addition to a game |
| Learning construction | 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. |
| Learning demo | 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. |
| Learning reflection | 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:   * Ease to understand * Fun factor * Use of Sphero * Creativity   Class discussion:   * Was the game a mix of challenge and fun? * What sort of features can students think of to enhance the game? * What was difficult about creating this game? * What are the challenges about coding this game for the Sphero * What are the benefits of using functions in this case? * Was this game easy to test/debug? * What were the challenges when designing the functions? * What were some of the advantages? * What other things do you need to consider when writing a program with functions? * Was it difficult to program the Sphero using functions? * What are the advantages and disadvantages of using functions? * What was challenging about using functions? |

**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 peer­assessment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **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 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 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. 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 F­6 Digital Technologies: Extended

* Unit 1 ­ Fundamental thinking skills

Year 7­8 Next Steps

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

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. Only a few are currently compatible with Sphero SPRK+

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

[SPRK Lightning Lab](https://medium.com/sphero-sprk/introducing-sprk-lightning-lab-ace9c7a8ecec) ­ Programming for Sphero Robots by Orbotix Inc. (Sprk+)

[Sphero](https://itunes.apple.com/au/app/sphero/id468699619?mt=8) by Sphero Inc. ( Sprk+)

[Sphero Exile](https://itunes.apple.com/au/app/sphero-exile/id553865533?mt=8) by Sphero, Inc.

[Sphero ColorGrab](https://itunes.apple.com/au/app/sphero-colorgrab/id553867540?mt=8) by Sphero Inc.

[Sphero Edu](https://itunes.apple.com/au/app/sphero-edu/id1017847674?mt=8) by Sphero Inc.

Digital Technologies Hub: <http://www.digitaltechnologieshub.edu.au>

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



Author: Celia Coffa This work is licensed under a Creative Commons Attribution­-NonCommercial 4.0 International License. Computer Science Education Research (CSER) Group, The University of Adelaide.