**Data handling**

**Lesson 3: Sensor gadget design**

**Introduction**

In this lesson, students consider the need for sensors to continually check for changes, developing their understanding through the use of unplugged activities and by writing algorithms using repetition and selection. They then apply their knowledge and understanding to design a gadget using a sensor and selection and evaluate their work against the set design criteria.

**Time:** @60 minutes

**Learning objectives**

* To explain how repetition is used when programming sensors
* To follow design criteria to design a product
* To write algorithms that show how sensors will be used

**Materials needed:** printouts of *Gadget with sensor planning sheet (***slide 14**), printouts of *Gadget with sensors evaluation sheet* (**slide 16**) printouts of *Algorithm support sheet*, lesson presentation.

**Lesson summary**

1. Introduction: Reading and writing algorithms (10 minutes)
2. Street lights (15 minutes)
3. Getting inventive (25 minutes)
4. Presenting your design (10 minutes)

**Introduction: Reading and writing algorithms (10 minutes)**

* Before starting the lesson, it would be useful to have a method of making the class darker if this cannot be done simply by turning lights off, closing blinds etc.
* Use **slide 3** to display an algorithm. Invite students’ suggestions on what the algorithm is instructing them to do and when they will do each action. Turn the lights on and off to make the classroom dark and light and get students to carry out the actions in the algorithms on **slide 3 and 4**.
* Display the structure of the algorithm and ask students to write a simple algorithm that uses the level of light as a condition (**slide 5**).

**Street lights (20 minutes)**

* Watch the time-lapse video of a street light turning on and off. Explain to students that this can be seen as an example of **selection**. Invite suggestions on what condition needs to be met and what to do if it is or isn’t met. Focus on responses that indicate the condition as being linked to the light level (is it dark?*)* and the actions are to turn the light on or off.
* Show **slide 7** and remind students of previous algorithms that they have written using decision boxes (revisit how the structure is used). Invite students to suggest what could go in each section. A completed example is included on **slide 8**.
* Invite a student to role play the street light by giving them a torch and asking them to follow the algorithm on **slide 9**. Close the curtains and/or turn off class lights to create a dark environment and reverse to create a light environment. It is likely that the student will continue to respond to the changes and turn the torch on and off accordingly.
* Explain to the students that the algorithm only instructs the user to check if it is dark once, therefore the action should only be carried out once. When using sensors, we have to program them to check the conditions constantly so they respond at the required time. Invite suggestions on how they have previously instructed computers to do things more than once (through the use of **repetition**).
* Display an alternative algorithm for being a street light and invite students to identify the similarities and the differences between this algorithm and the decision box-based algorithm on **slide 10** (see slide notes for answers).
* Invite students’ ideas on what the instruction ‘forever’ means - keep doing it - and why it is required in the algorithm (so the sensor is constantly checking if it is dark).
* Repeat the algorithms from the lesson introduction (**slides 11 and 12**), but this time see if students take note of the lack of ‘forever’ on **slide 12**. For this example, they should carry out the action the first time the light is changed and not again because they were only asked to check the light level once.

**Getting inventive (25 minutes)**

* Show **slide 13** to students and explain that they are going to design a gadget that can that either responds to changes in light level or temperature. Discuss how their design should be a representation of the gadget’s main features and purpose. Ask students to recall the term used in computing whereby the main information is focused on and extra detail is ignored (**abstraction**).
* Examples could include toys that light up when it gets dark, a glass that keeps a soft drink at the same temperature, a book that starts glowing when the lights are turned off so it can be read in the dark, and socks with heat pads that come on when the temperature goes below a certain level.
* As well as creating a labelled drawing of their design, highlight they need to write an algorithm to explain how their gadget will make use of selection and sensors.
* Students are not going to make these designs so should not be restricted by their understanding of the micro:bit. Instead they should concentrate on designing a gadget that makes use of a sensor to select which action to carry out.
* Give students copies of the *gadget with sensors planning sheet* (**slide 14**) and time to design their gadget and write the algorithm to explain how it will make use of sensors and selection.

**Presenting your design (10 minutes)**

* Organise the students into small groups (3-5). Give each student the opportunity to present their design to the rest of the group and explain where they have met the requirements of the design criteria (**slide 15**).
* Give out copies of the *Gadget with sensors evaluation sheets* (**slide 16**) and ask students to complete them to comment on the design and the algorithm.
* Discuss groups’ learning as a class, reviewing the learning objectives on **slide 17**.

**Extension ideas**

Students could design a persuasive poster to advertise their gadget.

**Differentiation**

**Support**

Students use the *algorithm support sheet* to help with structuring their algorithm.

**Stretch & challenge**

Students can be encouraged to create more sophisticated gadgets, including more than one sensor (or even designing their own sensors) and include these in their algorithms.

**Opportunities for assessment**

* Informal assessment of students’ understanding of how sensors work from class and group discussions.
* More formal assessment of students’ designs and algorithms.