**Nature art**

**Lesson 3: Digital nature representations**

**Introduction**

In this lesson, pupils are introduced to the micro:bit and how images can be represented by using the LEDs. They create visual algorithms to plan simple images before writing programs using the MakeCode editor to create their images.

**Time:** @60 minutes

**Materials needed:** lesson presentation, printouts of *Items to Represent with LEDs* and *LED Planner* handouts*,* laptops/computers, access to [the MakeCode editor](https://makecode.microbit.org/#editor), physical micro:bits (optional).

**Learning objectives**

* To understand that LEDs can be used to create image representations.
* To plan LED image representations.
* To construct programs to display LED image representations with the micro:bit.

**Lesson summary**

* Introduction: micro:bit nature art (5 minutes)
* Representing images with LEDs (10 minutes)
* Planning LED images (10 minutes)
* Programming LED representations (10 minutes)
* Paired programming (15 minutes)
* Evaluating images (10 Minutes)

**Introduction: micro:bit nature art (5 minutes)**

* Explain to pupils that they are going to view some representations of nature created using micro:bit.
* Invite pupils to remind you what logical reasoning is and ask them to predict what plants/animals the images you are about to show them created on the micro:bit may represent.
* Show pupils the images on **slide 3** and allow them to discuss their ideas with a partner before taking feedback. Guide pupils towards justifying why they think a specific plant/animal has been represented.
* Use **slide 4** to invite pupils to look at the images again and identify anything that all the images have in common. Invite predictions on how they think the representations have been created.

**Representing images with LEDs (10 minutes)**

* Use **slide 5** to introduce pupils to the micro:bit if needed, or as a recap.
* Focus on the set of LEDs (light emitting diodes) on the micro:bit. Explain that these are an example of an **output** (**slide 6**) and that can be controlled (turned off or on) using code.
* Ask pupils to discuss and share how they think the images seen on the previous slides were created (e.g. which LEDs are switched on/off?)
* Open the [simple images 1 program](https://makecode.microbit.org/#pub:_97wWYsARaetH) in the MakeCode editor and view the simulator in full screen mode. Run the program and ask pupils to identify what images are being represented, using the restart button to re-run the program. You can also use the MakeCode hex file provided.
* When taking feedback, ask pupils to identify what it was about the image that made them think it was a representation of that item (remind them by making predictions using the available evidence they are using logical reasoning).
* Show pupils the LED planner (**slide 7**) and invite suggestions on how this could be used to plan which LEDs would need to be turned on or off to represent a given image. Highlight that this is a simple visual algorithm that can be used for someone to program a micro:bit.
* Using **slide 8**, model as a class how to use the planner to create an image of a sad face. Ask pupils what parts of a face would be most likely convince someone else that it was a representation of a sad face. Explain that they are, as in lesson 1, using **abstraction** to highlight the most important information and disregard the parts that are not required.
* While recording, shade in a *wrong* LED and explain to pupils that by using pencils to plan the images changes can be made by rubbing out selected LEDs. Invite suggestions, on the name of the process by which errors are found and fixed as they did with their algorithms in lesson 2 (**debugging**).

**Planning LED images (10 minutes)**

* Give pupils a copy of the ‘Items to represent with LEDs’ and LED planner sheets and ask them to use them to plan their representations of at least one of the items on the list.

**Programming LED representations (10 minutes)**

* Recap the term ‘program’ (**slide 9**) and ask pupils to share their experience of programming so far.
* Explain that they are going to construct a simple program to show one (or more, if time allows) of their images on the micro:bit.
* Display **slide 10** to introduce pupils to the [MakeCode](https://makecode.microbit.org/#editor) editor before opening the webpage and asking pupils to discuss the links between the program and the output on the simulator.
* Invite pupils (either as a class, or if more confident, in pairs), to program a different image by modifying the existing program. If a mistake is made - selecting a wrong LED - ask pupils to suggest how this problem can be overcome.

**Paired programming (15 minutes)**

* In pairs, or small groups give pupils time to tinker with the MakeCode editor and construct programs to show the images that they created using the LED planner (**slide 11** can be used to introduce paired programming).
* If possible, pupils should snip/screenshot an image of the simulator displaying their image and save this to a shared area on the school network or to presentation software, so you can later show the images to the rest of the class from your computer.

**Evaluating images (10 Minutes)**

* Invite pairs/small groups to show the images they created to another pair/group or the rest of the class (**slide 12**). Ask the other pupils to share what they think the images represent.
* Ask pairs to discuss their programs with another pair, highlighting any mistakes they made whilst programming and how they overcame these (**slide 13**).
* Use **slide 14** to review the learning outcomes of the lesson and ask pupils to think/pair/share their main learning points from the lesson.

**Extension ideas:**

* Pupils could compare the MakeCode language with other graphical programming languages they have used. Similarities may include blocks that connect together and blocks having different colours. Differences may include programs organised vertically instead of horizontally, and different words used on the blocks.

**Differentiation**

**Support:**

* The first four (red, 1-4) items on the images to represent with LEDs sheet are items that can be represented with simple images. Pupils could be provided with images of the items they are representing.

**Stretch & challenge:**

* The final four (purple, 9-12) items on the ‘images to represent with LEDs’ sheet provide an additional challenge due to the complexity of the image to be represented. Pupils will need to abstract the most important detail and ignore the other details in order to be able to create a representation. They could be challenged further to see how many different representations (abstracting different detail) they can create for a given item.

**Opportunities for assessment:**

* Informal observations of pupils’ understanding of how the LEDs are used to create images through class discussions.
* Informal observations of pupils understanding of how to use the MakeCode editor to write their program.
* More formal assessment if wished of the pupils’ plans for their image representations and code.