**Digital flashcards**

**Lesson 2: Abstraction and programming**

**Introduction**

In this lesson pupils recap their learning from the ‘Nature art’ unit before being introduced to using the BBC micro:bit as a digital flashcard. They identify appropriate language vocabulary to represent, and plan and use the MakeCode editor to write sequenced-programs to displays LED images of these words. They will use and build on this program in subsequent lessons.

**Time:** @60 minutes

**Materials needed:** computers/laptops to access the MakeCode editor, large sheets of paper to record vocabulary list, printouts of the *LED planner (slide 8)*, example hex file (if needed).

**Learning objectives**

* To use abstraction when planning LED images
* To write programs that create LED images
* To sequence programs

**Lesson summary**

* Recapping LED images (10 minutes)
* Planning flashcards (25 minutes)
* Programming flashcards (20 minutes)
* Evaluating (5 minutes)

**Introduction: Recapping LED images (10 minutes)**

* Invite pupils to recap the previous lesson and ask pupils to demonstrate how to use the MakeCode editor to write programs to represent images using the micro:bit’s LEDs (**slide 3** - recapping work completed in the ‘**Nature art**’ unit).
* **Use slides 4 & 5** to give pupils images to represent. If helpful, challenge them to represent the images within a given time (one minute).
* Discuss the LED images pupils have created and their use of abstraction (**slide 6**).

**Planning flashcards (25 minutes)**

* Explain to pupils that they are going to use the MakeCode editor to create digital flashcards using a micro:bit. Invite suggestions on how a micro:bit could be used as a flashcard (**slide 7**).
* As a class, create a list of 10 nouns that could be represented using the micro:bit’s LEDs. Judge the suitability of suggestion by asking pupils if they would be able to represent the word using the micro:bit’s LEDs.
* Give out the LED planner (**slide 8**) and ask pupils to recall how they have used it previously.
* Ask pupils to select four words from the class list and plan their sequence of images using the LED planner sheet (**slide 9**).

**Programming flashcards (25 minutes)**

* Use **slide 10** to explain to pupils they should now program their images using the MakeCode editor.
* Recap how in the ‘**Nature art’** unit they wrote simple programs that displayed one image and that in this program the images need to be placed in a sequence. Ask pupils to discuss what the term **sequencing** means and how they have used it previously in their computing work (**slide 11**).
* Give pupils time to work in pairs to turn their algorithm into a program using the MakeCode editor, reminding them to test and debug as they go. An example of the type of program pupils could write is contained within the lesson downloads ([*DigitalFlashcard1*](https://makecode.microbit.org/#pub:_Ye9W9wdPPCEj)) (you may need to rearrange the comment blocks to see the code).
* Pupils will need to use these programs in the next lesson. They can access these by naming the program and using the same login details and the same computer or saving in a shared drive. Alternatively, they can anonymously share (publish) their program in MakeCode and copy the address (URL) to a text document from which the program can be opened.

**Identifying issues (5 minutes)**

* Advise pupils that there is a problem with the flashcard programs - can they identify what it is? (The images are displayed too quickly, with no delay between them).
* Invite pupils to run their programs and ask them to respond to the flashcards. Invite comments as to the problem with the flashcards (there is no time to respond). Highlight that they will be using their problem-solving skills to address this in the next lesson.
* Use **slide 12** to recap how sequencing has been used in the lesson and **slide 13** to revisit the learning objectives if you wish.

**Extension ideas:**

* Pupils could plan and program LED images to represent a song they have learnt in their studied language i.e. *Frère Jacques*.

**Differentiation**

**Support:**

* The algorithm could be constructed as a shared activity led by an adult. Two words could be represented as a group and then a further word represented independently.
* Pupils could be asked to represent at least two words in their sequenced program. Some pupils may benefit from having the blocks already sequenced so they only have to select the LEDs to be turned on in each image.
* For EAL pupils the MakeCode editor language could be changed (click on cog > Language).

**Stretch & challenge:**

* Pupils could be challenged to add more sequenced images in their plan and in their program.
* The MakeCode editor language could be changed to the target language being learnt (click on cog > Language).

**Opportunities for assessment:**

* Informal observations of pupils’ understanding of writing and sequencing programs.
* More formal assessment of pupils’ programs.