**Digital flashcards**

**Lesson 5: Debugging & evaluating**

**Introduction**

In this final lesson, pupils use their algorithms to program the BBC micro:bit as a digital number flashcard that meets design criteria. Upon completion, pupils evaluate their programs against the design criteria, consolidate and review the all computing concepts used in this unit.

**Time:** @60 minutes

**Materials needed:** Algorithms created in the previous lesson, laptops / computers to access the MakeCode editor, micro:bits if you have them, evaluation and evaluation support sheets for pupils, example hex file.

**Learning objectives**

* To follow an algorithm accurately to create a digital number flashcard
* To write and debug programs that meets design criteria
* To evaluate against design criteria

**Lesson summary**

* Copy my image (10 minutes)
* Recapping algorithms (5 minutes)
* Programming number flashcards (30 minutes)
* Evaluating and reviewing (15 minutes)

**Introduction: Copy my image (10 minutes)**

* Invite pupils to think/pair/share the ways found to program LEDs through tinkering in the previous lesson.
* Show the images in **slides 4, 5 and 6**. In pairs, ask pupils to write simple programs to create an identical output using the best block for the task: **show LEDs**, **show number**, or **show string**. Links to solutions are provided in the slides and in supplied HEX files.

**Recapping algorithms (5 minutes)**

* Invite pupils to explain how, in the previous lesson, they wrote an algorithm using the LED planner to show how they will program micro:bit to be a digital flashcard.
* Use **slide 8** to remind pupils of the design criteria for their product.
* Remind pupils of the importance of testing and debugging regularly as they construct their programs (**slide 9**).

**Programming number flashcards (30 minutes)**

* In pairs or small groups, give pupils time to follow their algorithms to program their digital flashcards using the MakeCode editor, reminding them to test and debug as they go.
* If using physical micro:bits, pupils should also transfer their programs to the device and test and debug them.
* An example of a program (*[DigitalNumberFlashcards](https://makecode.microbit.org/" \l "pub:_8m11eEEgaLma)*[)](https://makecode.microbit.org/" \l "pub:_8m11eEEgaLma) that meets the design criteria has been included in the lesson downloads.

**Evaluating and reviewing (15 minutes)**

* Display **slide 10** and ask pupils to reflect on their successes at addressing each point of the success criteria. Ask pupils to complete the evaluation sheet individually.
* If programs have been transferred to micro:bits, you may wish to allow pupils to swap and try out the digital flashcards.
* Review the unit as a whole by discussing understanding of the key computing concepts using **slides 11-15** and the lesson objectives on **slide 17** if you wish.

**Extension ideas:**

* Pupils could create digital flashcards that could be used by another class to support specific vocabulary being learnt in foreign language lessons or to develop subject specific vocabulary in another area of the curriculum.
* Pupils could use the flashcards as a memory game and even do a research project on whether numerical, graphical or written numbers are quicker to read, and which are easier to remember.

**Differentiation**

**Support:**

* Pupils use evaluation support sheet which could be completed as an adult led activity.
* For EAL pupils the MakeCode editor language could be changed (click on cog > Language).

**Stretch & challenge:**

* Pupils could swap digital flashcards and evaluate this as a ‘user.’ Are all aspects of the design criteria met? Can suggestions for further improvements be made (these should not be limited by their knowledge of programming the micro:bit)?

**Opportunities for assessment:**

* Informal observations of pupils’ understanding of algorithms and evaluations through paired, group and whole class activities.
* More formal assessment of pupils’ programs and evaluation sheets.