**Digital flashcards**

**Lesson 4: Predicting & experimenting**

**Introduction**

In this lesson, pupils develop their logical reasoning skills by matching programs with their outputs before experimenting (‘tinkering’) with the MakeCode editor to find additional ways of controlling the BBC micro:bit’s LEDs. They are introduced to design criteria for the use of micro:bit as a digital number flashcard and create an algorithm that meets these needs.

**Time:** @60 minutes

**Materials needed:** Computers / laptops with access to MakeCode editor, printouts of slides 4-6, printouts of *LED Planner*, sheets of A4 paper to write algorithms on, highlighter pens (yellow, green, blue, pink).

**Learning objectives**

* To use logical reasoning to identify the output of a program
* To tinker (experiment) to develop understanding
* To create an algorithm that meets given criteria

**Lesson summary**

* What’s my program? (10 minutes)
* Tinkering with LEDs (20 minutes)
* Designing a number flashcard
* Reviewing algorithms (10 minutes)

**Introduction: What’s my program? (10 minutes)**

* Give pupils print out of **slides 4 - 6**. In pairs or small groups, ask pupils to annotate the programs to predict the output.
* Use the links on each slide to run the programs using the micro:bit simulator (in full screen mode) – HEX files are also supplied.
* Ask pupils to use logical reasoning to identify which program would have created the sequence observed. When taking feedback, ensure pupils justify their choices by referring to the sequence of images and the length of delays to help to develop their logical reasoning skills.

**Tinkering with LEDs (20 minutes)**

* Explain to pupils that the LED images can be programmed using other blocks and ask them to tinker with the MakeCode editor to find other ways of controlling the LEDs (use **slide 8** if helpful to explain the concept of tinkering and why it’s important in Computing).
* Allow sufficient time for pupils to explore other ways of programming LEDs before inviting pupils to share their findings. Using **slide 10**, discuss which is the most effective program for creating a representation of the digit eight.
* Invite suggestions on how the results of their tinkering could be used to create another flashcard that would help some practice numbers to ten in their studied language.

**Designing a number flashcard (20 mins)**

* Share and discuss the design criteria for a number flashcard with the pupils using **slide 12**.
* Discuss how this could be planned as an algorithm using the LED planner (an example is included on **slides 13 & 14** of the lesson presentation).
* In pairs or small groups give pupils time to plan how they will use the micro:bit as a digital flashcard to practice numbers to ten the chosen language.

**Reviewing algorithms (10 minutes)**

* Use **slide 15** to display the design criteria for the flashcards and ask pupils to work in their pairs or small groups to highlight where on their algorithm they have met the different aspects of the design criteria.

**Extension ideas:**

* Pupils could develop an instruction leaflet that tells someone the different ways to create LED images using micro:bit.

**Differentiation**

**Support:**

* Pupils could use modified design criteria that require representations using the *show LED* blocks only.
* For EAL pupils the MakeCode editor language could be changed (click on cog > Language).

**Stretch & challenge:**

* Pupils could be challenged to create more representations in their algorithm.

**Opportunities for assessment:**

* Informal observations of pupils’ use of logical reasoning when making matching programs to outputs and tinkering when exploring additional ways to programs LEDs.
* More formal assessment of pupils’ algorithms and how it meets the design criteria.