**Digital flashcards**

**Lesson 3: Patterns & delays**

**Introduction**

In this lesson, pupils explore solutions to the problem identified in the previous lesson. They develop their understanding of the ‘wait’ command by using it in algorithms and micro:bit MakeCode programs when solving the problem. Pupils are introduced to the role of patterns in computing and identify patterns in their algorithms and programs.

**Time:** @60 minutes

**Materials needed:** Computers / laptops with access to MakeCode editor, *LED planners* from previous lesson, programs written in previous lesson, an example of a pupil’s algorithm written in lesson 1, micro:bits if you have them, example hex file.

**Learning objectives**

* To identify solutions to problems
* To identify patterns
* To use delays in algorithms and programs

**Lesson summary**

* Solving problems (10 minutes)
* Using delays with algorithms (15 minutes)
* Programming with delays (25 minutes)
* Reviewing improvements (10 minutes)

**Introduction: Solving problems (10 minutes)**

* Ask pupils to recall the problem with the digital flashcards they identified at the end of the previous lesson - the images move on without allowing sufficient thinking time.
* Display an example of an algorithm produced in lesson one and ask pupils to discuss with a partner why the person following this algorithm does allow thinking time.
* Identify that waiting for the person to respond would not be possible with the micro:bit but that ‘wait’ could be used in a different way. Ask pupils to think/pair/share ideas on how ‘wait’ could be used to display the images for longer and allow thinking time.

**Using delays with algorithms (15 minutes)**

* Model how to annotate the LED planner used in the previous lesson to identify the delays that need to be added and give pupils suitable time to use their LED plans from the previous lesson to create an algorithm that indicates the length of time each image will be shown for.
* Use **slide 7** to explain the role of patterns in computing and invite suggestions on how pupils have used patterns in their algorithm when deciding on the wait time given to each image.

**Programming with delays (25 minutes)**

* In their pairs, invite pupils to explore the blocks in the ‘basic’ menu in the MakeCode editor to find a block that can be used to delay the program and test out how it can be used. Note the ‘pause’ block uses **milliseconds**, so a delay of 2 seconds will be **pause (ms) 2000**.
* As a class, share pupils’ findings of which block to use and how it can be used.
* Ask pupils to modify their programs created in the previous lesson in line with the changes made to their algorithms.
* Remind them of the need to test and debug their programs as they go.
* If you have access to micro:bits, pupils should transfer them to the device once they feel their program is working and then try out their flashcard.
* An example of a program ([*DigitialFlashcard2*](https://makecode.microbit.org/#pub:_CX0g2XR8E2vT)) using pause block to create thinking time is included in the lesson downloads (you may need to rearrange the comment blocks to see the code).

**Reviewing improvements (10 minutes)**

* Display **slide 10** to the pupils and ask them to discuss their understanding of the computing concepts: evaluation, algorithms, patterns and debugging.
* Invite suggestions from pupils, on how they made use of each of these concepts to improve their digital flashcard
* Ask pupils for examples of how their work in other lesson could benefit from the process they undertook.

**Extension ideas:**

* Pupils could write an algorithm for having a short conversation in the language you are studying e.g. saying hello and asking someone’s name.
* If you have micro:bits, pupils could construct an algorithm that instructs others how to transfer a program from the MakeCode editor to the micro:bit.

**Differentiation**

**Support:**

* Pupils could write algorithms and modify programs as part of an adult led group.
* For EAL pupils the MakeCode editor language could be changed (click on cog > Language).

**Stretch & challenge:**

* Pupils could explore simple sentence patterns in the language you are studying and write an algorithm to help someone follow one of the patterns.

**Opportunities for assessment:**

* Informal observations of pupils’ solutions, use of patterns and programs.
* More formal assessment of pupils’ use of delays in their programs.