**Sensory classroom**

**Lesson 2: Light patterns**

**Introduction**

In this lesson students focus on creating a light pattern for a sensory aid by writing a pseudocode algorithm using iteration to meet a set criteria and programming their algorithm using micro:bit.

**Time:** @60 minutes

**Materials needed:** Lesson plan, lesson guide, A3 paper, light pattern planning sheet (printed from slide 10), access to [MakeCode website](https://makecode.microbit.org/), micro:bits (optional), USB connectors and battery packs.

**Learning objectives**

* To use pseudocode to write an algorithm for a light pattern
* To use iteration in algorithms and programs to create a repeating light pattern
* To evaluate an algorithm and program to ensure they meet criteria

**Lesson summary**

1. Introduction: A sensory classroom (10 minutes)
2. Creating repeating patterns (10 minutes)
3. Writing algorithms (10 minutes)
4. Turning algorithms into programs (15 minutes)
5. Evaluating programs (10 minutes)
6. Wrap up (5 minutes)

**Introduction: A sensory classroom (10 minutes)**

* Ask students to recall from the previous lessons why sensory rooms are used in schools and other settings. Explain that you would like to bring sensory elements into your classroom and briefly recap the potential benefits for some students (**slide 3**).
* Give out A3 paper and use **slide 4** to set students a ‘*300 second challenge’* to produce a sketch detailing how a standard classroom could be converted into a sensory classroom. Invite students to share their ideas with the rest of the class.

**Creating repeating patterns (10 minutes)**

* Explain to students that they are going to create a sensory aid for a classroom that uses light patterns and discuss which patterns would be most appropriate for a sensory aid and why (**slides 5-7**).
* Encourage students to identify the commonalities between the images selected non-threatening, symmetrical and make reference to identifying ‘patterns’ in Computational Thinking if you wish.
* Show pupils **slide 8** and invite suggestions on how the image could be used to plan and record a light pattern before giving out blank copies of the light pattern sheet (**slide 10**) and asking pairs to create a repeating light pattern, using the diagrams to record each stage of their pattern (**slide 9**).
* Give students appropriate time to design and record their repeating patterns, labelling each diagram to show the sequence of their light pattern.
* Invite pairs to share their patterns with another, explaining why they have chosen the specific images sequence and what is displayed at the end of the sequence.

**Writing algorithms (10 minutes)**

* Use **slide 11** to recap algorithms with students and explain they are now going to write a pseudocode algorithm to show how their repeating pattern could be displayed on a microbit.
* Share the criteria and highlight the concepts they will be using (**slides 12-15**).
* When students finish their algorithm, invite them to compare, test and debug with another pair, checking their algorithms meet the criteria (**slide 16**).

**Turning algorithms into programs (15 minutes)**

* Ask students to briefly recap their micro:bit experience and access the [MakeCode editor](https://makecode.microbit.org) (**slide 17**).
* Using paired-programming (**slide 18**), give students time to create their program, reminding them to test and debug regularly. If necessary, provide additional programming support according to your students’ levels of confidence and experience.
* Once students have a working program, ask them to investigate different ways the program could be started (i.e. how the user can interact with the sensory aid). This will help them at a later stage and could include pressing buttons A or B (separately or together); shaking the micro:bit; rotating or tilting micro:bit; dropping micro:bit - the last two inputs can only be achieved with physical micro:bits.
* If you have physical micro:bits, students can connect their micro:bit and transfer the program (if possible with a connected battery supply).

**Evaluating programs (10 minutes)**

* Revisit the program criteria (**slide 19**).
* If you have time, show students a program that has some elements of the given criteria but not all (see supplied [**criteria review hex file**](https://makecode.microbit.org/#pub:_0KU0P98eAKTE)). Invite students to identify what parts of the criteria have been met and to suggest how the program can be debugged or adapted to meet it.
* Ask students to show their programs to each other and identify how they have met some/all of the given criteria, allowing time to revise their program if needed.

**Wrap up (5 minutes)**

* Ask students to demonstrate the variety of different ways they used to start their micro:bit program with another pair (**slide 20**), and share 2 things they have learnt in this lesson. Revisit the learning objectives on **slide 21** if you wish.

**Extension ideas:**

* Students could act out their algorithms by arranging themselves into a 5 x 5 array and turn a piece paper over to show if the light is on or off and/ or create an animation to represent a bubble tube (see [**bubble tube example hex file**](https://makecode.microbit.org/#pub:_aAWWU44xk9z7)).

**Differentiation**

**Support:**

* Students could use the **algorithm support sheet**to construct instructions for their algorithm.
* Students could use the ‘show icon’ block from the ‘basic’ menu in the MakeCode editor which provides a selection of pre-designed images which can be used to create a sequence of different images.

**Stretch & challenge:**

* Students could write a program that use more than one input to activate different repeating patterns (see [**repeating pattern example project hex file**](https://makecode.microbit.org/#pub:_4iHaxRTpzRAb)), or that uses variables which would randomly select a repeating pattern (see [**random image pattern example project hex file**](https://makecode.microbit.org/#pub:_3bzLfcer2UxU)).

**Opportunities for assessment:**

* Informal observation of students’ during activities and discussion, especially evaluative comments.
* Informal, or more formal assessment if wished, of students’ light patterns, algorithms and programs.