**Musical micro:bit**

**Lesson 4: Controlling music with inputs**

**Introduction**

In this lesson, pupils explore further the inputs that the BBC micro:bit can use to trigger events. They read programs and make predictions about when outputs will occur. They experiment with selection by using conditions that make use of the micro:bit’s accelerometer so that all the notes in an octave can be played using inputs. After writing, testing and debugging their program, they consider the potential benefits of using a micro:bit for people who find playing musical instruments difficult.

**Time:** @60 minutes

**Learning objectives**

* To identify how inputs are used in programs
* To write programs that use inputs and selection
* To write and evaluate algorithms

**Materials needed:** lesson presentation, laptops/computers with access to MakeCode editor, printouts of *selection in a musical scale* sheet and *selection in a musical scale (challenge)* sheet, **starter**hexfile, **instrument-support** hex file, **instrument** hex file, large sheets of paper, paper to write algorithms on, optional micro:bit devices, battery packs, USB leads, crocodile clips, headphones/speakers

**Lesson summary**

1. Introductory activity: 3, 2, 1, program (10 minutes)
2. Selecting conditions (30 minutes)
3. Writing algorithms (15 minutes)

**Introduction: 3, 2, 1, program (10 minutes)**

* Show the [videos on **slide 3**](https://youtu.be/oPcdq0VAAh4) and on [**slide 4**](https://youtu.be/jPd3De6wnbo). After each video has been shown, challenge pupils to write a program to get micro:bit to play the same musical phrase. If you wish, pupils can use the [**starter hex file**](https://makecode.microbit.org/#pub:_3J38yA7EXXmY) which contains some of the blocks needed for the programs. If you have access to physical devices, you may wish to rehearse transferring the program to the device and connecting micro:bits to headphones/speakers to play the phrase.
* After pupils have programed the first musical phrase, discuss the techniques they used. Ask pupils if anyone managed to write the program without getting extra blocks from the menu. Some pupils may have duplicated the blocks already given and therefore saved time. Invite a pupil, who knows how to do this, to model to the rest of the class how to duplicate a block (right-click on the block and select duplicate). Invite pupils to test this out when programing the musical phrase from **slide 4**.

**Selecting conditions (35 minutes)**

* Show pupils the **Instrument** MakeCode program on **slide 5** and ask them to use their existing knowledge to make statements about the program (when a certain condition is met, a certain note is played). Ask pupils to work with a partner to write down their statements on a large sheet of paper, before inviting them to feedback ideas to the class. Remind pupils they are using logical reasoning (see speaker notes for example responses).
* Use the [link](https://makecode.microbit.org/_CJw8f8fXoMoX) in the slide to open the same program in the MakeCode editor. Invite pupils to use the simulator to show how to get a micro:bit to play certain notes (pressing button A will produce low G note, pressing button B will produce low A note, shaking the micro:bit will produce low B). Remind pupils that when a computer carries out an action only when a condition is met is an example of selection.
* Focus on the block that contains the condition ‘is shake gesture’ and explain to pupils that the micro:bit uses the **accelerometer** to check if the device has been shaken.
* Click on the drop-down menu of the block to show other accelerometer-based inputs:



* Explain to pupils that they are going to finish this program by adding the rest of the notes in the octave (middle C, middle D, middle E, middle F & middle G). Invite pupils’ suggestions on what blocks would need to be added to program a micro:bit to play the next note. Explain that they are going to use the accelerometer-based inputs that involve moving the micro:bit (the first seven options). Select an appropriate condition and then add a block to specify which note will be played.
* Explain that the simulator cannot represent these inputs so it will have to be tested by downloading and transferring the program to a physical micro:bit which should be connected to speakers/headphones in order to play the music. Invite a pupil to demonstrate how to do this if needed. If you do not have access to the hardware the program can still be completed using the MakeCode editor.
* Give pupils a copy of **instrument hex file** and ask them to work with a partner to complete the program so that the micro:bit has a different output for each note in the octave. Use **slides 6 & 7** to explain to pupils how to complete the program and record the inputs on the worksheet to state which condition they used for each note. You may wish to change the octave to suit the pitched instruments you have used as a class.

**Writing algorithms (15 minutes)**

* Once pupils have programed their micro:bits to play the notes in the chosen octave when certain conditions are met, use **slide 8** to explain that they are going to write a simple algorithm to allow someone to play one of their musical phrases from lesson one using the micro:bit’s inputs.
* Give pupils time to work with their partner to write an algorithm that instructs someone which micro:bit inputs to use to get the device to play a musical phrase. Pupils should move on to this task when they have finished the program. If you do not have access to physical micro:bit hardware, the task can still be completed but pupils are creating an algorithm for when they have access to a micro:bit.
* Once pupils have written an algorithm, ask them to swap algorithms with another pair and evaluate this algorithm. Use the questions on **slide 8** to support their evaluation and feedback.
* If you wish, use **slide 9** to revisit the learning outcomes of the session.

**Extension ideas**

* Pupils could use the *start melody* block from the music menu to create melodies that are played when certain inputs are used. Then they can compose music by sequencing the different melodies and playing them through micro:bits attached to speakers or headphones.

**Differentiation**

**Support:** Pupils could use the **instrument-support hex file** to support the structure of their program and their block choices.

**Stretch & challenge:** Pupils could program from a blank canvas and select all the inputs that are used as conditions for when a note is played. Pupils could also use the **selection in a musical scale (challenge)** sheet to record their conditions.

**Opportunities for assessment:**

* Informal assessments of pupils’ understanding of inputs, selection, algorithms, programming and evaluations through whole class and paired activities.
* More formal assessment of pupils algorithms and programs if you wish.