|  |  |  |  |
| --- | --- | --- | --- |
| **Making Music with the BBC micro:bit** | | | |
|  |  | |  |
| Designing and programming a device that can be used as a musical instrument in a class performance | | | |
| **Subject(s):** Design and Technology, Computing, Engineering  **Approx time:** 80-120 minutes |  | | **Key words / Topics:**   * BBC micro:bit * blocks * design brief * metronome * performing music * programming * rhythm, harmony, melody * sound |
| **Stay safe**  Whether you are a scientist researching a new medicine or an engineer solving climate change, safety always comes first. An adult must always be around and supervising when doing this activity. You are responsible for:    • ensuring that any equipment used for this activity is in good working condition  • behaving sensibly and following any safety instructions so as not to hurt or injure yourself or others    Please note that in the absence of any negligence or other breach of duty by us, this activity is carried out at your own risk. It is important to take extra care at the stages marked with this symbol: ⚠ | | | |
| **Suggested Learning Outcomes** |  | |  |
| * To understand that a simple piece of music is made up of rhythm, harmony and melody parts. * To be able to design a programmable musical instrument that produces one of the three parts to a piece of music. * To perform the musical part that they have created with their device as part of a wider class performance. | | | |
| **Introduction** |  | |  |
| This is one of a series of resources to support the use of the BBC micro:bit in Primary Design & Technology, Computing and Engineering lessons. This resource focusses on learners designing, programming and using a programmable device that can be used as a musical instrument in a class performance. | | | |
| **Purpose of this activity**  In this activity learners will design and create a programmable device that can be used to play a rhythm, harmony and/or melody part in a class music performance. They will analyse a design brief and design criteria before taking one of two routes through the activity – designing and producing the programming themselves, or using pre-written programs that they can download straight onto their devices and/or edit as they go through.  This activity could be used as a main lesson activity to teach about how programmable devices can be used to generate music within D&T, Computing and/or Engineering. It could also be used as part of wider STEM-based scheme of learning focussed on how programming can be used to embed intelligence into products and systems. | | | |
|  |  | |  |
| **Activity** |  | | **Teacher notes** |
| **Introduction (5-10 minutes)**  Teacher to introduce the activity and safety notes. Teacher to explain that learners will be producing a programmable device that will act as a musical instrument in a class performance.  **Starter activity (10-20 minutes)**  Teacher to show the starter program to learners. Ask them to look at the code and explain what they think it does. Finally, get them to simulate and then run the code on their micro:bit to see what it actually does.  **Design context, brief and criteria (5-10 minutes)**  Explain to learners that the design context is what sets the scene for the project that they will be working on. Share and discuss the context given on presentation slide 4. Then discuss the design brief shown in slide 5 and the design criteria on slide 6. Teacher to hand out the resources and equipment required.  **Producing the programmable device (40-50 minutes depending on route taken) ⚠**  Learners can be taken through this activity in one of two different ways. Route 1 (presentation slide 7) or route 2 (slide 8). Ensure and demonstrate safe practices when working with electronic devices e.g. ensure polarity of battery leads are correct, do not touch exposed wires, remove power if components are overheating etc.  Route 1  Learners to follow the instructions on slide 7 to open a new project. They should then write their own code to satisfy the needs of the design brief and the musical part they have been asked to produce (rhythm, harmony or melody). The teacher could show or provide access to the relevant example programs to help them, or take them through how to write the code step by step.  Route 2  Learners to follow the instructions on slide 8 to open the relevant example program on screen and download to their micro:bit. The main focus in this instance will be on the use of their device to produce their musical parts.  **Class performance (20-30 minutes)**  Learners to use their programmed devices to produce their musical parts in the whole class performance of the full piece of music. |  | | This activity could be undertaken firstly in small groups, with each group responsible for generating a rhythm, melody or harmony part. Then, everybody could come together as a class for the final performance, combining all of their parts together.  **Micro:bit versions**  This resource works on both V1 and V2 of the BBC micro:bit. With V1 an external speaker will be required for sound. There are also some other limitations with using a V1 micro:bit (e.g. no logo touch), so a V2 is the preferred option for this activity.  **Starter**  Teacher could deliver this starter using the PR of the PRIMM (Predict-Run-Investigate-Modify-Make) process:   * Predict – show the code and ask students what it does. * Run – enter the code and run it in the simulator and then on the micro:bit and observe what it actually does.   This is a simple input/output program; when you shake the micro:bit (input) it plays a short note (output); this leads-in to the general principle of all other parts of this  activity, i.e. that you can use an input of any type to trigger a musical output. If the triggering and playback responds quickly enough, this can be used as part of a musical performance as a home-made instrument.  **Design brief and criteria**  Learners could be split into three groups.   * Group 1 will create the rhythm parts of the performance. * Group 2 will create the harmony parts. * Group 3 will create the melodies.   It would also make sense for group leaders to be appointed, so each team can communicate with each other to ensure they are creating parts that would work with each other. Or the teacher could coordinate this.  **Producing the programmable device**  Learners can be taken through this in one of two ways:   * Route option 1 – Learners themselves create the code required to meet the design brief, with help as required from the teacher. In this instance the main focus is the programming activity. * Route option 2 – All of the example programs are treated as pre-written and downloaded onto the micro:bits ‘as is’. In this instance the main focus is on the overall system and its use in context to create the musical parts.   **Class performance**  Learners may need to practice their different pieces in groups (e.g. the rhythm group will perform their rhythm part together first), then all groups will need to come together as a class for the class performance. They may need extra time and assistance to ensure all their parts fit well together and to ‘tweak’ them accordingly.  **Example programs**   * Starter example program - This program is for the starter activity (presentation slide 3). * Drum:bit example program - This is the simplest of the three main programs to be written by learners. It builds on the starter activity and uses the pre-defined sounds of the sound synthesiser. It can be difficult to see which sound is selected in the sound synthesiser, but by clicking on the waveform and then pressing the gallery button, you can choose from a selection of percussive sounds.   Teachers could present the table of suggested input triggers from a table, and then get learners to choose an input trigger and an output sound, and connect them together in the code editor. All of the suggested input triggers have simple one-block event handlers, and playing a sound is one block of code. Teachers might also choose to get learners to build multiple triggers and sounds into their drum:bit (as we do in the worked example), so that they can build a complete multi-part drum-kit.  Note that many of the event handlers trigger on-release rather than on-press, so teachers might need to make sure learners know about this to prevent delays in sound production.   * Harmony:bit example program - This can be used to generate 3 notes that naturally belong together in a musical piece; often this is called the '3 chord trick'. For example A, D and E are notes that can be used for a 3-chord trick. The micro:bit cannot play more than one note at a time, so it will only play the root notes of these three chords, but that is sufficient to harmonise with. The program uses gesture control so the micro:bit could be attached to a glove, shoe or hat. * Melody:bit example program - This plays different programmed parts in a multi-bar musical piece, usually called part A, part B, and optionally part C. Different combinations of parts A B and C give different song structures. In this code, learners write their own 2 or 3 musical bar parts, and can then perform them in different orderings. In this example, button A plays part A, button B plays part B, and logo touch plays part C. The parts play once, to completion, when triggered.   There are a number of ways to build the parts of the melody – teachers could provide samples of the musical parts and get students to enter them, or they could describe some music theory and let them design their own; or a musically trained teacher would know how to compose simple pieces (opportunity to link with music lessons). The music grid is used as it is very visual and learners at this level will find it easy to use, although in this instance learners are limited to the key of C and a set 8/8 time signature. It does however provide a good starting point which can be expanded upon as desired.  If the limitation of being ‘stuck’ with the key of C needs to be escaped, teachers could get learners to use the Play Note blocks instead. They can then instruct them as to which notes appear in which key signature, and they can then compose or transcribe music in any key.   * Metronome:bit example program - The metronome is provided code, and allows teachers to 'synchronise’ the playing of multiple instruments to a common beat. It starts at 120 beats per minute and a 4/4 time signature, which are the most common settings for that age group. It plays a high 'pip' on the beat at the start of the bar, and a low 'boop' for other beats in the bar, helping students to keep in sync with the beat and the bar structure of a performance. On the first beat of the bar, the metronome:bit shows a large heart icon, and on the other beats in a bar it shows a small heart icon. Thus there is also a visual clue for students as to when the bar starts.   The teacher can speed up or slow down the bpm, button A slows it down by 10 and button B speeds it up by 10. The display will show the first 2 digits of the bpm whenever you change it (so 120 displays as 12 and 60 displays as 06). The range of bpm is set as 20bpm and 360bpm.  The teacher can change the time signature of the metronome, which is especially useful if the class is trying to practice or perform an existing written piece of music notation with a specific time signature. Pressing the logo changes to the next time signature – this is important because the beat length and the number of beats in the bar is dictated by the time signature, and the 'pip' and 'boop' pattern will be created according to the selected time signature. So, while 4/4 an 2/2 are mathematically the same figure, musically they have a different sound (4/4 covers most traditional music, 2/2 is like a swing beat). The time signatures programmed are, in this order: 4/4 3/4 2/4 2/2 6/8, and when you get to the end and press the logo again it cycles back to 4/4. Each touch of the logo shows the time signature as a 2-digit number on the display.  Pressing the RESET button at the back will restore the metronome to 120bpm 4/4 time. Teachers could add new time signatures themselves by adding a pair of digits to the end of the TIMESIGS constant in the OnStart block, but note that single-digit numerators and denominators are only supported.  If showing this code to learners, use the actual program provided and zoom in on the different parts so they are clear for learners to see. |
| **Differentiation** |  | |  |
| **Basic** |  | | **Extension** |
| * Provide the example programs as set code that can be downloaded ‘as is’, or edited as required. * Produce the programs in sections, with the teacher demonstrating how to do these one or two blocks of code at a time. * Give learners partially completed programs for them to add the missing blocks of code. |  | | * Program a metronome that can be used to keep a common beat for the ‘instruments’. * As a class, create several different musical pieces that could form a ‘concert’ for parents. |
| **Resources** |  | | **Required files** icon-docicon-pdficon-ppt |
| * BBC micro:bits (V1 or V2) and associated USB download cables * 3 V power supplies for micro:bits (e.g. 2 x AAA battery packs) * External speakers for sound (if using V1 micro:bit) * Computers with access to the internet |  | | Teacher presentation – BBC Micro:bit Making Music  Example HEX programs:   * Starter example program * Drum:bit example program * Harmony:bit example program * Melody:bit example program * Mertronome:bit example program |
| **Additional websites** |  | |  |
| * **BBC Micro:bit homepage:** <https://makecode.microbit.org> * **How to upload HEX files to the program editor:** <https://support.microbit.org/support/solutions/articles/19000065686> * **BBC Bitesize – What is musical notation?** <https://www.bbc.co.uk/bitesize/topics/zcbkcj6/articles/z994dnb> * **YouTube – Learning about music for kids:** <https://www.youtube.com/watch?v=YjFIlLKjmkI> | | | |
| **Related activities (to build a full lesson)** |  | |  |
| **Starters** (Options)   * Use a mind map or spider chart to analyse the design brief and criteria. * Use the information on presentation slide 3 to analyse the starter program and how it works. | | **Plenary**   * Evaluate how well the device performed when used in the class performance. What went well and what could be improved? * Self/peer assess the completed device against the requirements of the design criteria. | |

|  |
| --- |
| **The Engineering Context** film |
| * Programmable systems are an integral part of the world we live in today. Almost all electronic devices, from smartphones to washing machines to complex aircraft control systems, rely on programmable devices for them to function. It is therefore vital for systems engineers to develop skills in using programming to embed intelligence into electronic systems. * There is a range of excellent career opportunities available in the sound and audio engineering industry. Understanding of how music is created and the technologies is used is a very useful starting point for this. * Music is an excellent context to explore how programmable systems can be used to solve engineering and social problems. |

|  |  |
| --- | --- |
| **Curriculum links** | |
| **England: National Curriculum**  Design & Technology KS2   * apply their understanding of computing to program, monitor and control their products.   Computing KS2   * design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts. * use sequence, selection, and repetition in programs; work with variables and various forms of input and output. | **Northern Ireland Curriculum**  The World Around us KS2   * Design and make models. * The effects of adding components to simple circuits. |
| **Scotland: Curriculum for Excellence**  Technologies   * TCH 2-09a, TCH 2-12a * TCH 2-13a, TCH 2-14a, TCH 2-15a | **Wales: National Curriculum**  Primary – Science and Technology   * Design thinking and engineering offer technical and creative ways to meet society’s needs and wants. * Computation is the foundation for our digital world. |

|  |  |  |
| --- | --- | --- |
| **Assessment opportunities** | | |
| * Formal teacher assessment of completed programs and systems. * Self/peer assessment of completed devices against the design criteria. * Formal assessment of class musical performance. * Informal assessment of practical skills used. | | |
|  |  |  |