SCRATCH LESSON PLAN

Make Some Noise: Exploring Sounds and Music in Scratch

In this lesson, we will explore the Scratch sound library, which is full of a variety of sound effects, voices, and longer sound loops. We will also try the music extension blocks that let you use the sounds of a variety of instruments and actual notes with customizable beat lengths to create digital instruments, songs, and more! Using sound in a game, animation, story, or interactive project can add depth and emotion. Let's explore some of the ways you can add sound to Scratch projects by using sounds from the Scratch sound library, recording your own sound, uploading a sound, or using music blocks. Along the way, we will also experiment with options to make your sound project interactive using inputs like keyboard keys, the mouse, video motion, or even the features of your face or the Makey Makey combined with items like bananas! You may want to make sure your class is prepared with headphones, but they are sure to have some fun making noise and making coding connections.

Audience:

Classroom Teachers, Instructional Technology Specialists, Library Media Specialists, Informal Learning Environments

Time: Approx 2.5 hours total
- Part 1: Exploring the Sound Blocks - 40-60 min
- Part 2: Exploring the Music Blocks - 60 min
- Part 3: Reflect and Share - 15-30 min

We have provided timing, but you may opt to spread out over more sessions if learners need/want more time to tinker or you want to take more time to brainstorm, plan, and reflect.

Objectives (Learners Will):
- Identify good use-cases for sound in Scratch projects
- Create projects using sound in a variety of ways
- Remix our starter projects to add personalized touches and additional elements
- Reflect on their finalized projects and creative process with peers
- Communicate and share their projects with their learning community

See page 12 for aligned standards.

Resources for Learners:

- [Sounds and Music Cards](#) (Student-Facing Coding Cards) - printable cards students can use to follow along with the lesson
- [Starter projects](#) to remix and explore:
  - [DJ Scratch Cat](#)
  - [Catch the Fish, Increase the Pitch](#)
  - [Piano](#)
  - [Musical Buttons using Video](#)
  - [Musical Droplets](#)
  - [Drum Sequencer](#)
  - [Fur Elise with Music and My Blocks](#)
  - [Sound Graph](#)
- [Sounds in Scratch](#) (Tutorial Video)

Additional resources provided throughout the guide.

Special Note About Uploading Sounds:

- You can choose a MP3 or WAV file.
- Please keep each of your files under 10MB.
- Do not upload materials under copyright.
- Uploads must follow the [Community Guidelines](#).

Part 1: Exploring the Sound Blocks

Choosing a Sound (10 minutes)

Music and sound in Scratch can be approached in multiple ways. First, let's explore using a sound from the sound library and ways to change/manipulate it.

Step 1: Click on a sprite or the backdrop and select the "Sounds" tab. Hover over the Sounds menu at the bottom of the tab, and select "**Choose a Sound**."

Step 2: Back on the Code tab, under the Sound category, there are a couple of block choices for adding sound. Drag a sound block to the code area. If you have added multiple sounds, select the one you want from the dropdown list. Then, **try placing the sound block at a point in your script**.

Step 3: **Explore the difference between "start sound"** (which starts the sound and then immediately continues to the next block) **and "play sound until done"** (which waits until the whole sound has finished before continuing to the next block). Try each in a "forever" loop or add another block, like a "say" block, after the sound block to see the difference.

Note: Optionally, to record an original sound, hover over the Sounds menu at the bottom of the tab, and select "**Record**." Or choose "**Upload Sound**" to upload a pre-recorded sound.
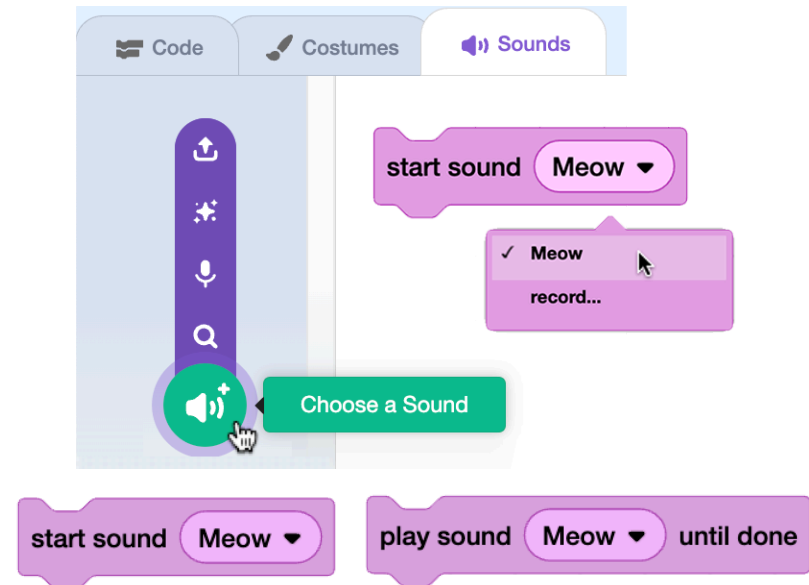
*Sound tab and "start sound" and "play sound until done" blocks.*
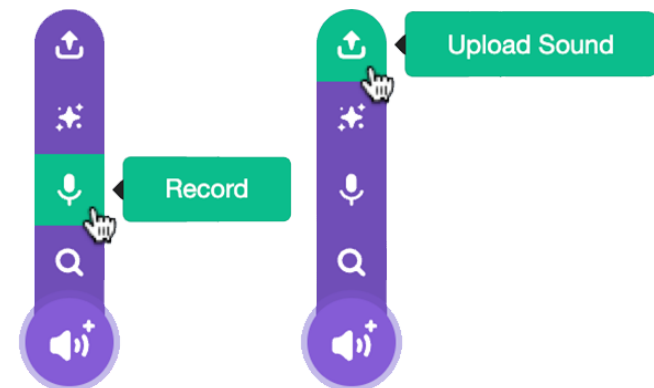
Sound menu "Record" and "Upload Sound" options.

Step 4: Once you have chosen a sound there are a few ways to customize a sound/manipulate it. One way is to use the **sound editor tools** to make edits, like shortening the length or adjusting the volume or adding an effect like fade in and out or reversing it.

Step 5: How might you use sounds in a project? One way is to **trigger sounds to play with keyboard keys or mouse clicks**. Check out our starter project "DJ Scratch Cat." Explore and remix this project to think about how to pair sounds or layer sounds to create something new.

Resources:
- Sounds and Music Cards (Student-Facing Coding Cards) - printable cards students can use to follow along
- DJ Scratch Cat (Starter Project) - to remix and explore
- Sounds in Scratch Tutorial Video and Written Guide
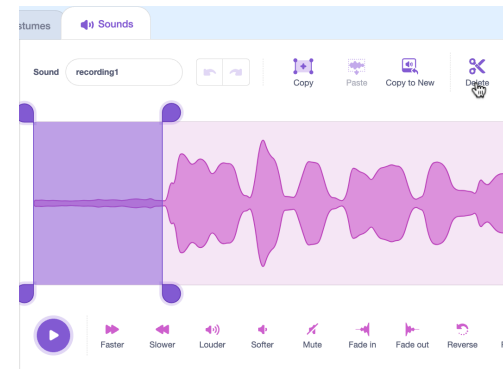

Sound editor tools window.
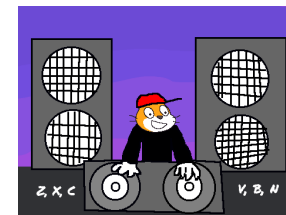
"DJ Scratch Cat" scratch.mit.edu/projects/11640429.

Pitch and Volume - Customize/Manipulate Sound via Code Blocks (30-50 minutes)

Another way to customize and manipulate sound in Scratch is via code blocks.

Step 1: Explore blocks in the Sounds category that can set or change the pitch or the volume. There are also blocks to stop all sounds currently playing or clear sound effects, like pitch.

Step 2: How might you use such blocks in a project? Take a moment to brainstorm ideas.
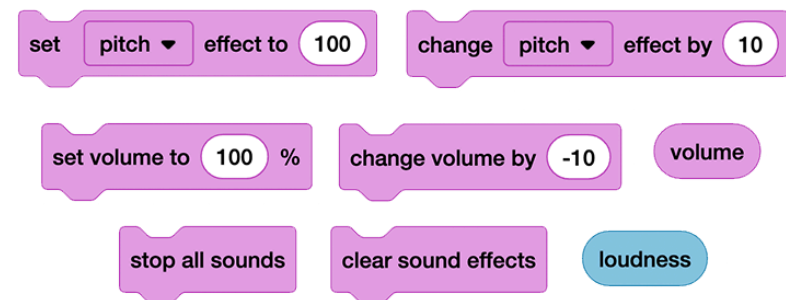

Some sound and loudness block options.

Step 3: One way might be giving users control over **hearing or muting continuous background sound** in a project. There are a number of ways to approach this. A few methods are shown in the project "Checking Status with Variables Example 1" or see another approach to the right. These solutions involve using a conditional statement. See resources below for more ideas.

Step 4: Another way to add emotion and excitement to a game could be **playing with the pitch of a sound**. Check out our starter project "Catch the Fish, Increase the Pitch." Explore and remix this project. Notice as you click on all 30 fish that the music gets higher in pitch creating a feeling of urgency. How can sound add an emotional component to a project?

You could also try a variant on our "Musical Droplets" starter project using pitch instead of musical notes.

Resources:
- Sounds and Music Cards (Student-Facing Coding Cards) - printable cards students can use to follow along
- Catch the Fish, Increase the Pitch (Starter Project) - to remix and explore
- How to use Pitch Effect (Project)
- Conditional Statements Part 1 and Part 2 (Video Tutorial)
- Conditional Statements Written Guide and Coding Cards
- Singing Sandwich (Example Project) - one example of playing with pitch by a Scratch Team member
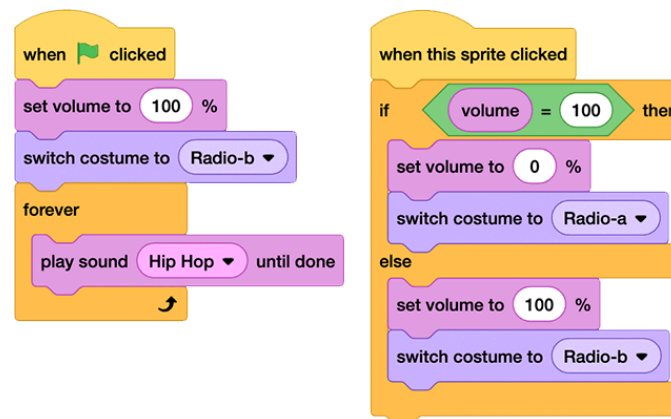
One example of how to start and stop continuous sound.
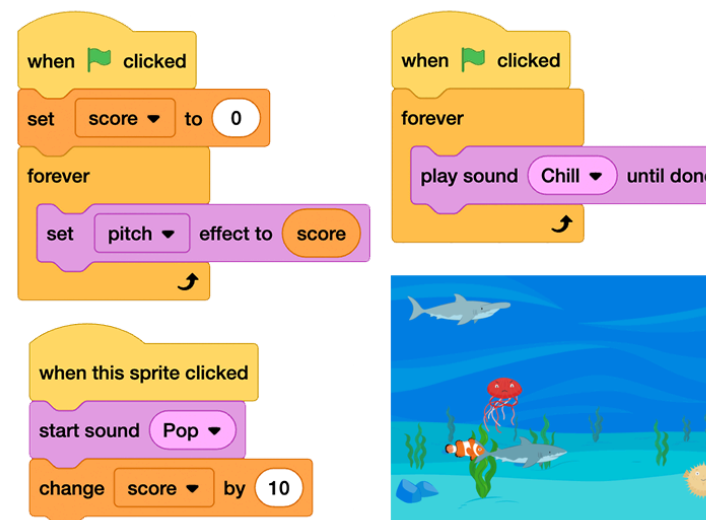
"Catch the Fish, Increase the Pitch"
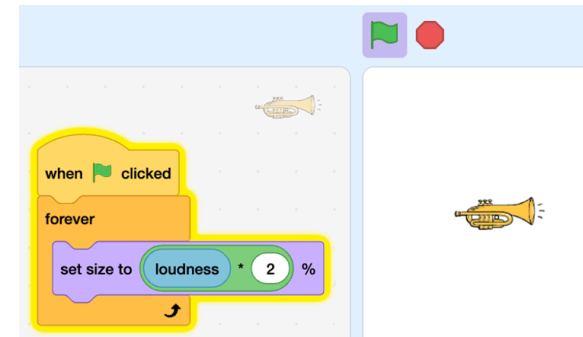*scratch.mit.edu/projects/1106268602.*

Alternative Options

Speaking of volume, did you know there is also a **"loudness" reporter block** under the Sensing category? This block records the "loudness" of the noise that a microphone receives, on a scale of 0 to 100, to control things in Scratch.
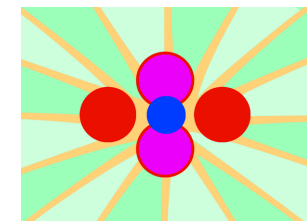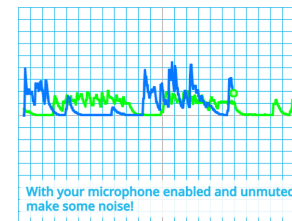
How could you use this in a project? Try using it in the "set size" block inside a "forever" loop. Then, make some noise!

Example script using "loudness" to control a sprite's size.

Or check out our starter projects "Sound Graph" and "SoundFlower." Enable your microphone in the browser (nothing will be recorded or stored). Then, explore these projects and see how singing, playing music, or just making fun noises at different volumes creates an effect. How might you create an interactive art piece to accompany a musical performance?

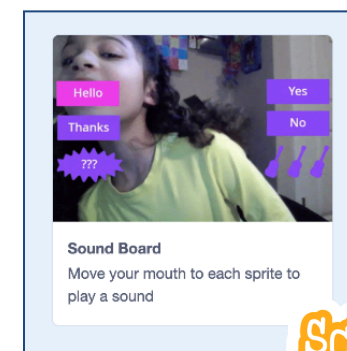*"Sound Graph" scratch.mit.edu/projects/1105532968 and "SoundFlower" scratch.mit.edu/projects/1111537402.*

You can also **use our experimental Face Sensing AI blocks to create a soundboard** or a musical instrument that can be played with the features of your face, like your nose. See our coding cards for more information.

Part 2: Exploring the Music Blocks

Try one or more of these options to start experimenting with the music extension blocks.

Option 1: Create a Digital Piano (30-60 minutes)

Add the Music extension by clicking on the extension menu in the lower-left corner of the project editor and choosing "Music."

Step 1: Check out our starter project "Piano." Explore and remix this project. We have set up a basic piano using the "play note _ for _ beats" block. Click in the note input bubble to see the piano keys that appear so that you can **choose a note** attached to a number.

Step 2: The beat or BPM (beats per minute) is a basic rhythmic unit of a measure. You can **make the beat faster or slower** by changing the number in that input bubble. Adjust using different numbers and test the difference.

Step 3: Try changing the instrument, either via the slider we have provided or by changing the script. Note that **changing the instrument** on one sprite does not change it for all sprites in a project, so you could have a separate instrument for each key. How does using a variable make it easier to adjust the instrument for all keys at once?

Step 4: In your remix, you could change what the piano looks like, add computer keyboard shortcuts, add a slider to change the length of the beat for all sprites at once, or add higher and lower notes.
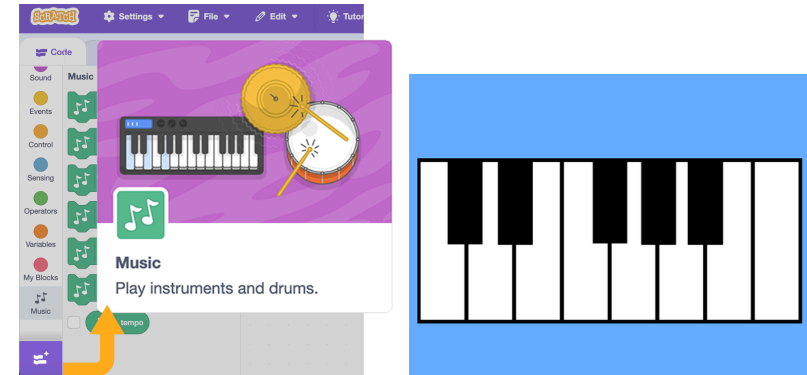
Scratch Extension Menu and "Piano" scratch.mit.edu/projects/1106245381.
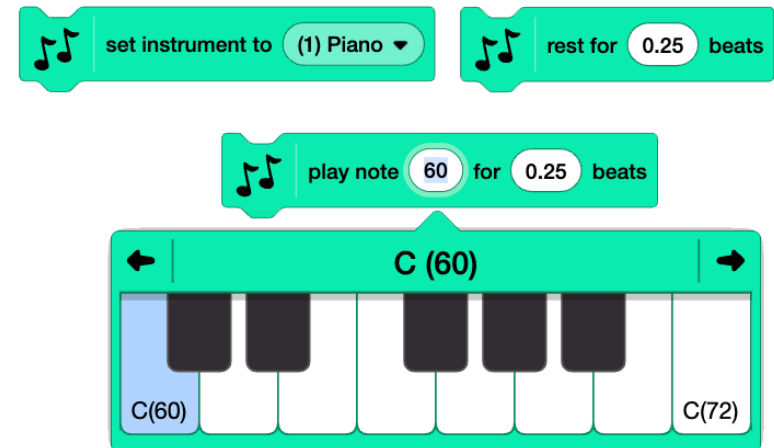
Music blocks to begin with "set instrument," "rest for _ beats," play note _ for _ beats."

What about **creating an alternate piano keyboard**? Check out our starter projects "Musical Buttons using Video" or "Musical Droplets" as examples of using video motion or the position of the mouse to play notes.

One thing to note about the "Musical Buttons using Video" project is that we have tried to **make it more accessible by adding visual effects** when notes play for those hard of hearing or deaf. How can you make your musical projects visual as well as audio so more people can experience them, or experience them in different ways?

Another way to extend this lesson is to **use a Makey Makey to turn your digital keyboard into a physical one** using conductive materials like bananas, foil, etc., as the physical keys. See our coding cards for more information.

Or check out our starter project "Drum Sequencer," an **interactive project where the user creates the beat**. The starter project uses sounds from the library, but you could experiment with additional drum sounds by using the "play drum _ for _ beats" block from the music extension.
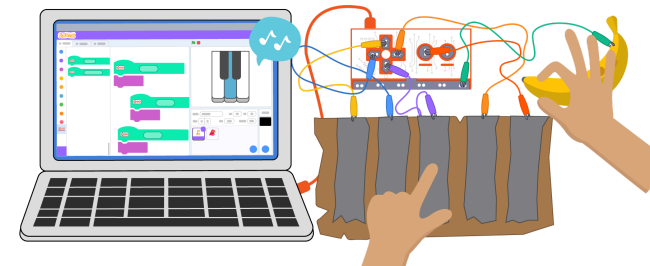
Resources:
- Sounds and Music Cards (Student-Facing Coding Cards) - printable cards students can use to follow along
- Piano, Musical Buttons using Video, Musical Droplets, and Drum Sequencer (Starter Projects) - to remix and explore
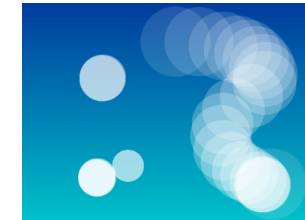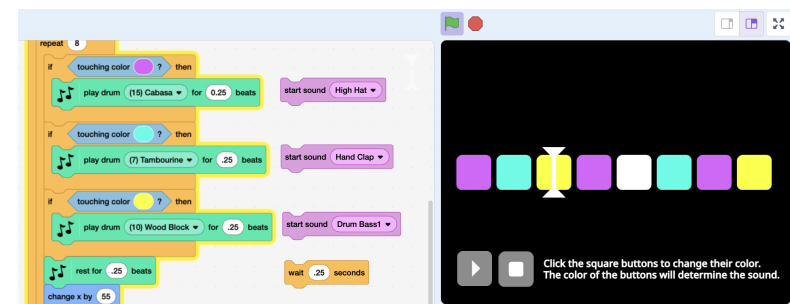

"Musical Buttons using Video"
scratch.mit.edu/projects/1105110383 and "Musical Droplets"
scratch.mit.edu/projects/1111576868.

*Create a physical piano by attaching notes to keyboard keys or using a Makey Makey and conductive objects.*

"Drum Sequencer" scratch.mit.edu/projects/1111562971.

## Option 2: Compose or Recreate a Song Using Music Blocks (30-60 minutes)

Think about the structure of a piece of music. Typically there are verses and choruses that are repeated throughout the song. You can use the music blocks to compose your own original composition or recreate a song!

Step 1: If re-creating a song, do some research to see if you can find a list of the notes you will need.

Step 2: When you have your song, identify any repeating parts.

Step 3: Are you creating a song with a simple melody, or are you creating chords and layering sounds or notes to play at the same time? There are different approaches you can try using sound blocks and instrument sounds from the library or music blocks. Experiment!

Step 4: Experiment with the tempo block, which controls how fast or slow music extension blocks play notes and drum beats. A standard tempo is 60 BPM, which means one beat will be played each second. What happens if you create a sequence of "play note" blocks and use the same beat but change the tempo?
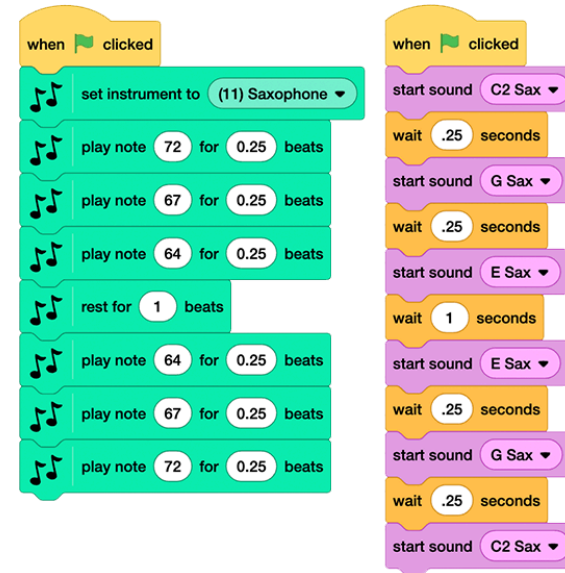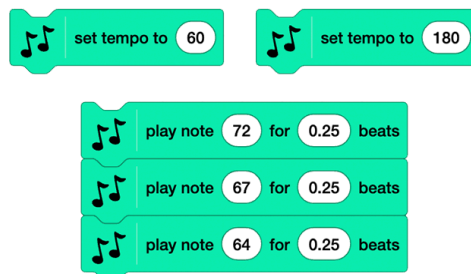
Example of two different ways to create a melody.
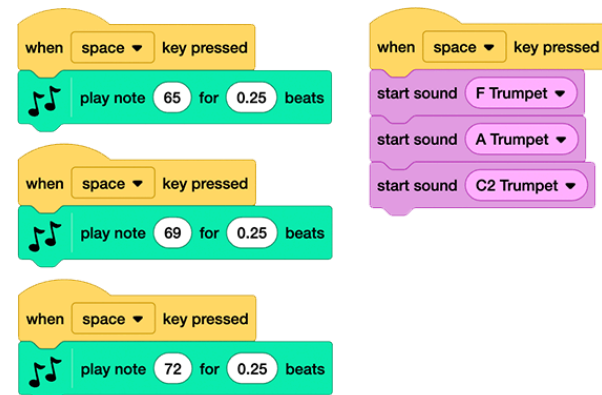
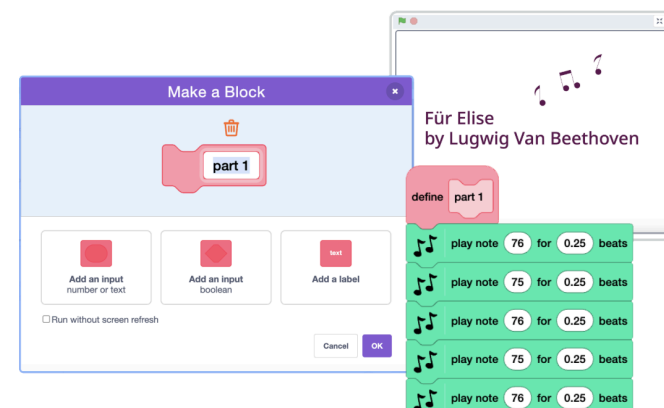*Example of two different ways to create a chord.*

Step 5: For repeating parts, rather than write the same sequence of notes over and over whenever they appear in your song, you can place those notes in a My Block and simply call that block each time you need it. For an example, check out our starter project "Fur Elise with Music and My Blocks." You can **create multiple My Blocks for different sections/parts**. My Blocks can also be placed within other My Blocks to further simplify the code. See our coding cards for more information.

"Fur Elise with Music and My Blocks"
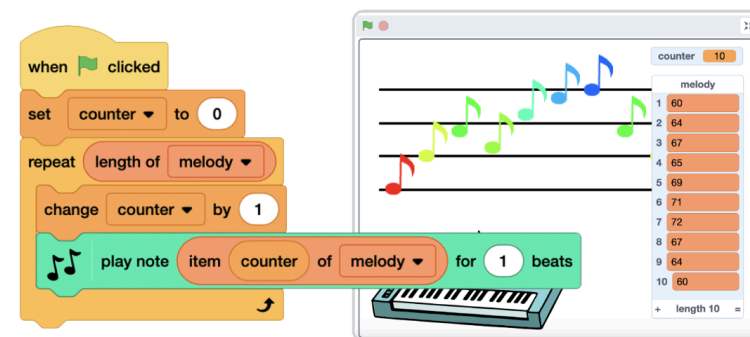*scratch.mit.edu/projects/1106259376.*

Step 6: Or try creating a melody project by **storing song notes in a list that creates a musical score** that can be played. See the example project "List Example 2 MelodyList remix" which repeats through a list of stored notes and stamps them on the screen for a visual effect. See our coding cards for more information.

Resources:
- Sounds and Music Cards (Student-Facing Coding Cards) - printable cards students can use to follow along
- Fur Elise with Music and My Blocks (Starter Project) - to remix and explore
- List Example 2 MelodyList remix (Example Project) - shows using a list to store notes
- My Blocks Part 1, Part 2, and Part 3 (Video Tutorial)
- My Blocks Written Guide and Coding Cards
- Variables and Lists Part 1 and Part 2 (Video Tutorial)
- Variables and Lists Written Guide and Coding Cards

"List Example 2 MelodyList remix"
scratch.mit.edu/projects/962308051/editor.

Part 3: Reflect and Share

Reflect (15-30 minutes)

Learners can reflect on their project creation and process as they complete the Sharing Sheet. Next, their peers are encouraged to leave feedback for the creator as they view the projects in a studio or participate in the gallery walk.

Resources:
- Show-and-Tell Sharing Sheet or Project Gallery Walk Self-Reflection and Peer Feedback Sheet (Worksheet)
- Scratch Design Journal (Worksheet) - This can be used to help students imagine, plan, iterate, and reflect throughout all of the phases of their project's development.

Share Option #1: Create a Class Studio to Gather Shared Projects

Studios are a space on Scratch where users can come together to make, share, and collect projects related to a particular theme, idea, or prompt. Set up a class studio* for your learners and add their original asset projects. Learners are encouraged to take time to look at projects and read/listen/interact with them to learn more about their peers.

Resources:
- Teacher Account Guide (Written Guide) - This resource contains information on setting up teacher accounts and student accounts, managing classes, and class studios.
- Scratch Studios Guide (Written Guide) - General information on setting up and managing.

Note: Learners need a Scratch account and access to the online editor to participate in this option.

Share Option #2: Gallery Walk

Have each participant's project open on their computer or other device. Participants can walk around a room, or take turns sharing their screen in a virtual space, to experience each other's creations. Or display one project at a time on a large screen. Learners are encouraged to take time to look at projects and read/listen/interact with them to learn more about their peers.

More Things to Try

- **Scratch Design Journal** (Worksheet) - This can be used to help students imagine, plan, iterate, and reflect throughout all of the phases of their project's development.
- **Debugging Strategies Posters** (Printable Posters) - These align with the debugging tips available in the editor.
- Have you seen the **Chrome Music Lab**? Our own Scratch Team member Eric R helped create it and it is a fun way to experiment with learning music via fun, hands-on experiments.

Looking to learn more? The Scratch Foundation has partnered with The Achievery to provide free beginner and intermediate creative coding lesson plans on a variety of topics for educators, caregivers, and learners. **Sign up (for free!) for the Achievery by using our custom partner code "SCRATCH"** when you register to support our work! https://www.theachievery.com/account/signup

See our units "Looperman," "Notation and Rhythm," "Coding a Digital Piano," and "Draw with Your Voice."

Standards Aligned

CSTA Standards	ISTE Standards	CASEL Framework	RITEC Indicators
Link to full standards	Link to full standards	Link to full standards	Link to full standards
1B-AP-08 Compare & refine algorithms 1B-AP-10 Create programs 1B-AP-11 Decompose problems 1B-AP-12 Modify, remix, or incorporate 1B-AP-14 Give appropriate attribution 1B-AP-15 Test and debug	1.1a Learning Goals 1.1d Technology Fundamentals 1.5.c Decompose Problems 1.5.d Algorithmic Thinking 1.6.b Creative Communicator	- Self-awareness - Self-management	- Autonomy - Competence - Creativity

This lesson also fulfills all three of the ISB Indicators of Playful Learning (Choice, Delight, Wonder), developed by the Pedagogy of Play (PoP) research project at Harvard University.

Tip: If you would like to translate this lesson plan, **click here to make a copy** of this Google doc.