Creating Procedural Music with MetaSounds

Learn how to create procedural music with MetaSounds.

Creating Procedural Music with MetaSounds

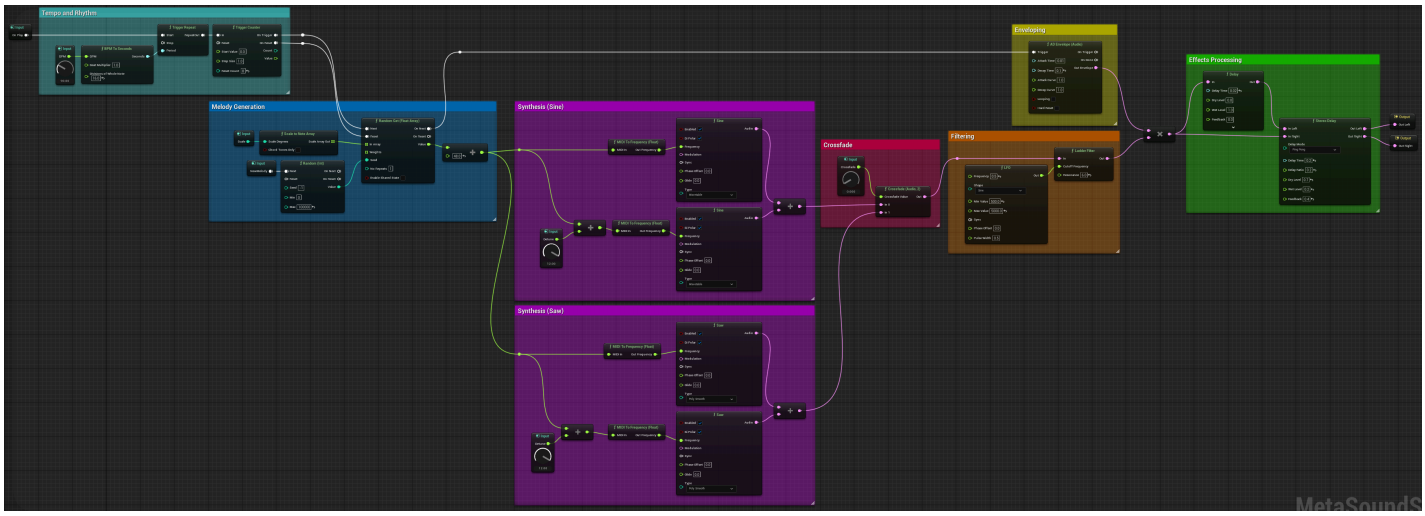
MetaSound is a high-performance audio system providing audio designers with complete control over a Digital Signal Processing (DSP) graph to generate sound sources.

This guide teaches you how to create a gameplay-driven procedural music system using a **MetaSound Source** and Blueprint.

Prerequisites

This guide requires a project with player character movement support, such as the First Person Template project.

1 - Create the Music MetaSound



Click image for full size.

Start by creating a MetaSound that plays procedural music. Follow each substep below to build the graph above, section by section.

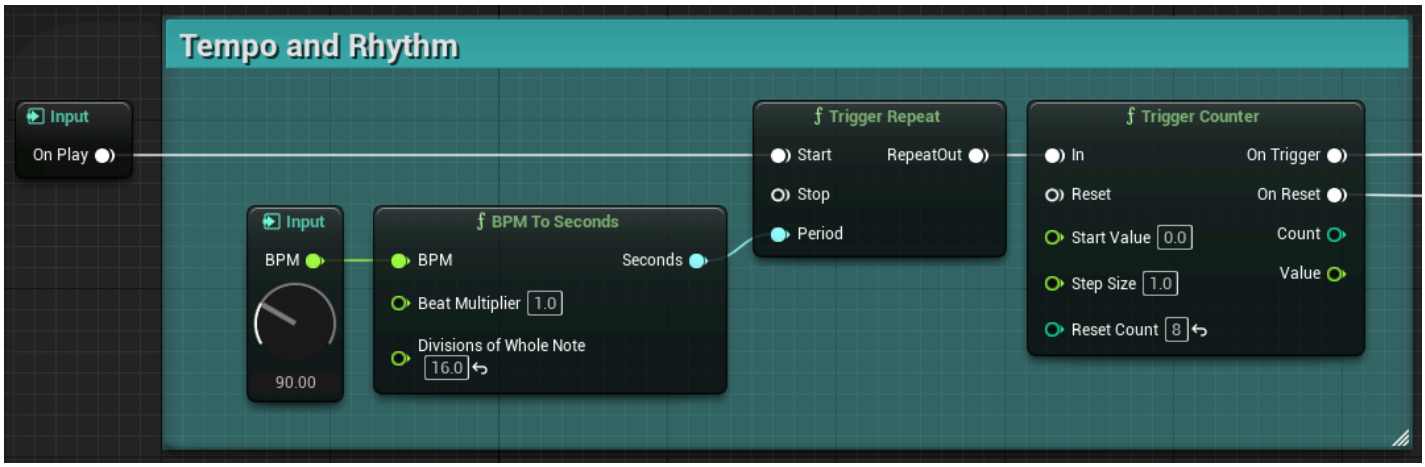
This process exposes several input parameters for access from Blueprint in future steps.

1.1 - Initial Setup

Create a MetaSound Source that supports persistent stereo audio.

1. Create a MetaSound Source.
    a. In the **Content Browser**, click the **Add** button.
    b. Select **Audio > MetaSound Source**.
    c. Give the newly created MetaSound a name, such as `MSS_Music`.
2. Double-click the MetaSound to open the **MetaSound Editor**.
3. In the **Interfaces** panel, click the **Remove (Trash Bin)** button next to the **UE.Source.OneShot** Interface entry. This removes the **On Finished Output** node, which isn't used on persistent sounds such as ambience or music.
4. Click the **MetaSound** button on the **MetaSound Editor Toolbar**.
5. In the **Details** panel, click the **MetaSound > Output Format** dropdown and select **Stereo**. This replaces the **Out Mono Output** node with an **Out Left** and an **Out Right Output** node.

1.2 - Tempo and Rhythm

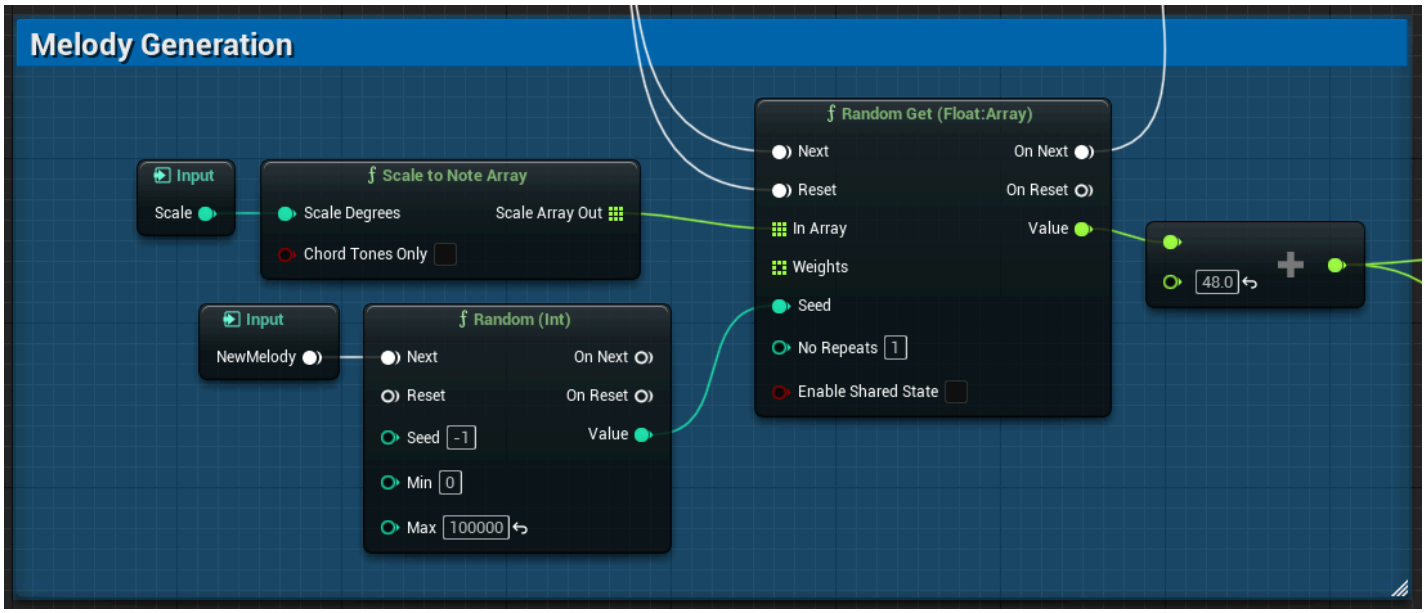


Build the Tempo and Rhythm section to control the timing of the music with Trigger nodes.

1. Find the **On Play Input** node and drag off of the pin into an empty space. Enter "Trigger Repeat" into the node search to create a connected node. You can move the node around the graph by dragging it.
2. On the **Trigger Repeat** node:
    a. Drag off the **Period** pin and create a **BPM To Seconds** node.
    b. Drag off the **RepeatOut** pin and create a **Trigger Counter** node.
3. On the **BPM to Seconds** node:
    a. Set **Divisions of Whole Note** to 16.
    b. Drag off the **BPM** pin and select **Promote to Graph Input**. This creates a **Float Input** node named BPM.
4. Select the **BPM** node.
5. In the **Details** panel, set **Default Value > Range** to 60.0, 180.0.
6. On the **Trigger Counter** node, set **Reset Count** to 8.
7. Select all of the nodes except the **On Play Input** node, right-click one of the selected nodes, and select **Create Comment From Selection**.
8. Name the comment box "Tempo and Rhythm."

> You can change the color of a selected comment box in the **Details** panel.

1.3 - Melody Generation

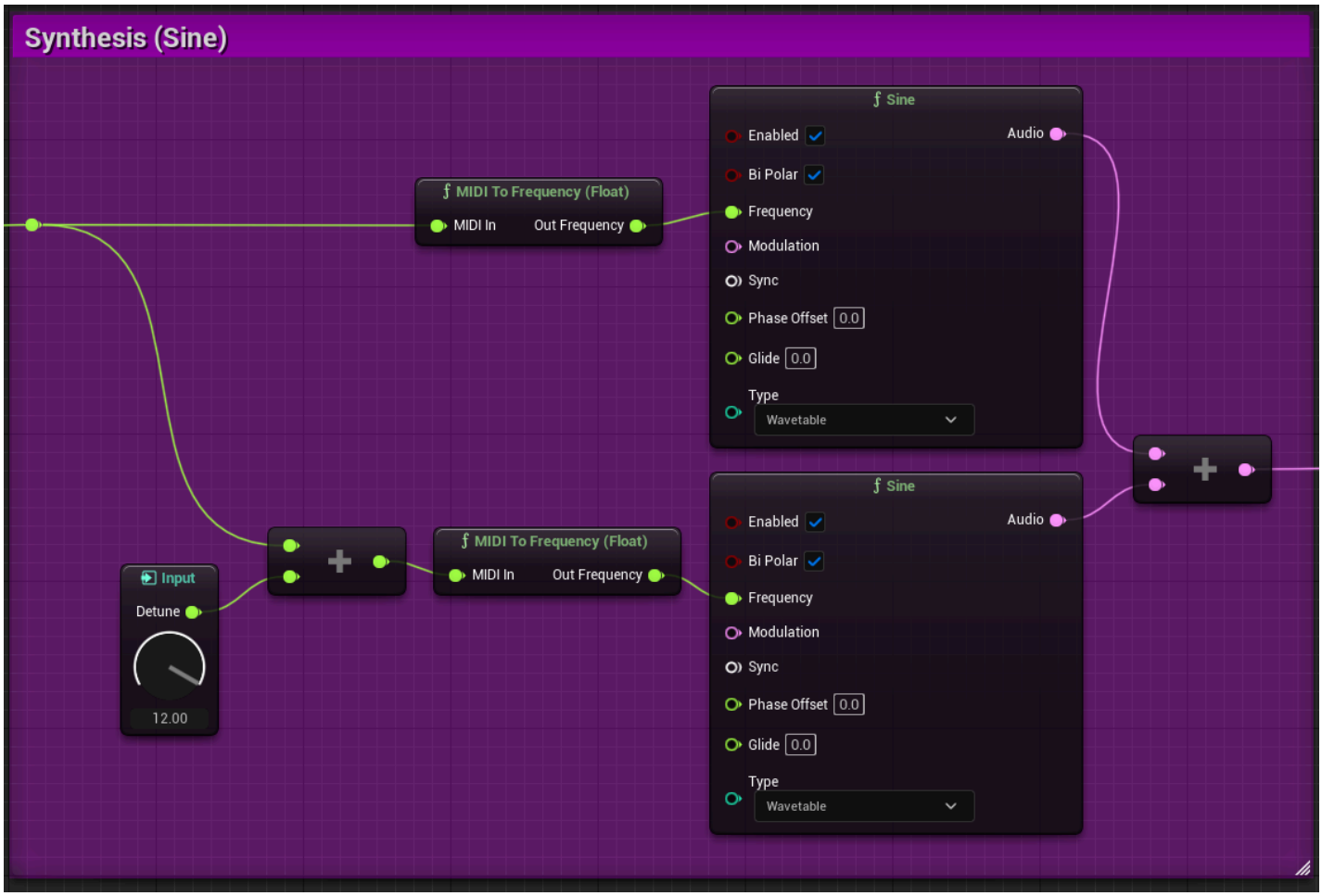


Build the Melody Generation section to produce a random melody in a specified scale.

1. On the **Trigger Counter** node in the Tempo and Rhythm section:
    a. Drag off the **On Trigger** pin and create a **Random Get (Float:Array)** node.
    b. Drag off the **On Reset** pin and connect it to **Reset** on the **Random Get (Float:Array)** node.
2. On the **Random Get (Float:Array)** node:
    a. Drag off the **In Array** pin and create a **Scale to Note Array** node.
    b. Drag off the **Seed** pin and create a **Random (Int)** node.
    c. Drag off the **Value** pin and create an **Add (Float)** node.
3. On the **Scale to Note Array** node, drag off the **Scale Degrees** pin and select **Promote to Graph Input**. This creates an **Enum Input** node named Scale Degrees.
4. Select the Scale Degrees node.
5. In the **Details** panel:
    a. Set **General > Input** to "Scale" to rename the input.
    b. (Optional) Set **Default Value > Default** to the music scale you want to use. This defaults to Major Scale.
6. On the **Random (Int)** node:
    a. Set **Max** to 100000.
    b. Drag off the **Next** pin and select **Promote to Graph Input**. This creates a **Trigger Input** node named Next.
7. Select the Next node.
8. In the **Details** panel, set **General > Input** to "NewMelody" to rename the input.

9. On the **Add (Float)** node, set the bottom addend to 48.0. This is an offset of four octaves.
10. Enclose the new nodes in a comment box named "Melody Generation."

1.4 - Synthesis (Sine)



Build the Synthesis (Sine) section to produce a sine wave at the frequency of the input note.

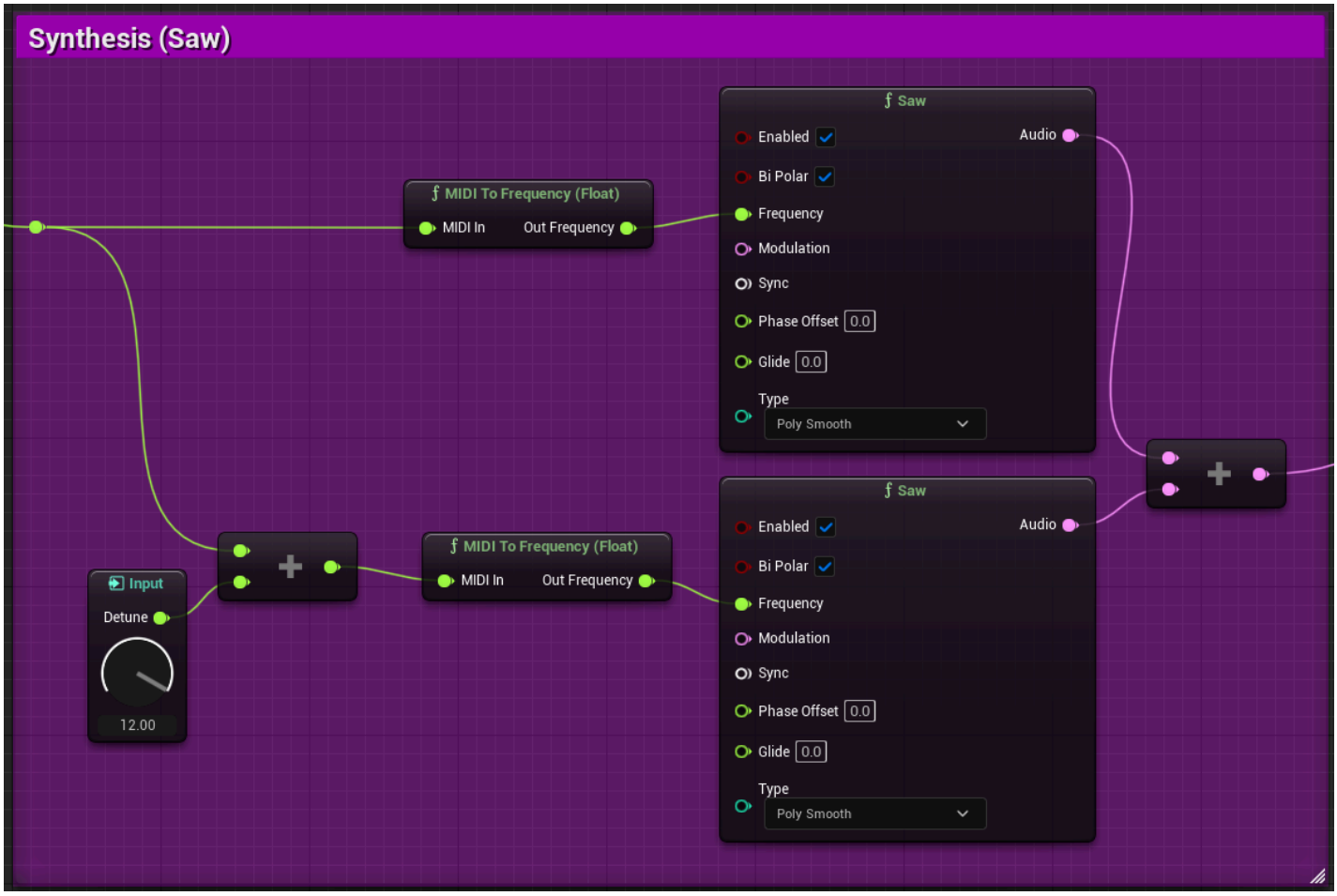
1. On the **Add (Float)** node in the Melody Generation section:
    a. Drag off the output pin and create a **MIDI To Frequency (Float)** node.
    b. Drag off the output pin again and create another **Add (Float)** node.
2. On the new **Add (Float)** node:
    a. Drag off the output pin and create another **MIDI To Frequency (Float)** node.
    b. Drag off the bottom addend pin and select **Promote to Graph Input**. This creates a **Float Input** node named AdditionalOperands.
3. Select the AdditionalOperands node.
4. In the **Details** panel:
    a. Set **General > Input** to "Detune" to rename the input.

b. Set **Default Value > Default** to 12.0.

c. Set **Default Value > Range** to 0.0, 12.0.

5. On each of the **MIDI To Frequency (Float)** nodes, drag off the **Out Frequency** pin and create a **Sine** node.
6. On the first **Sine** node, drag off the **Audio** pin and create an **Add (Audio)** node.
7. On the second **Sine** node, connect the **Audio** pin to the bottom addend of the **Add (Audio)** node.
8. Enclose the new nodes in a comment box named "Synthesis (Sine)."

1.5 - Synthesis (Saw)

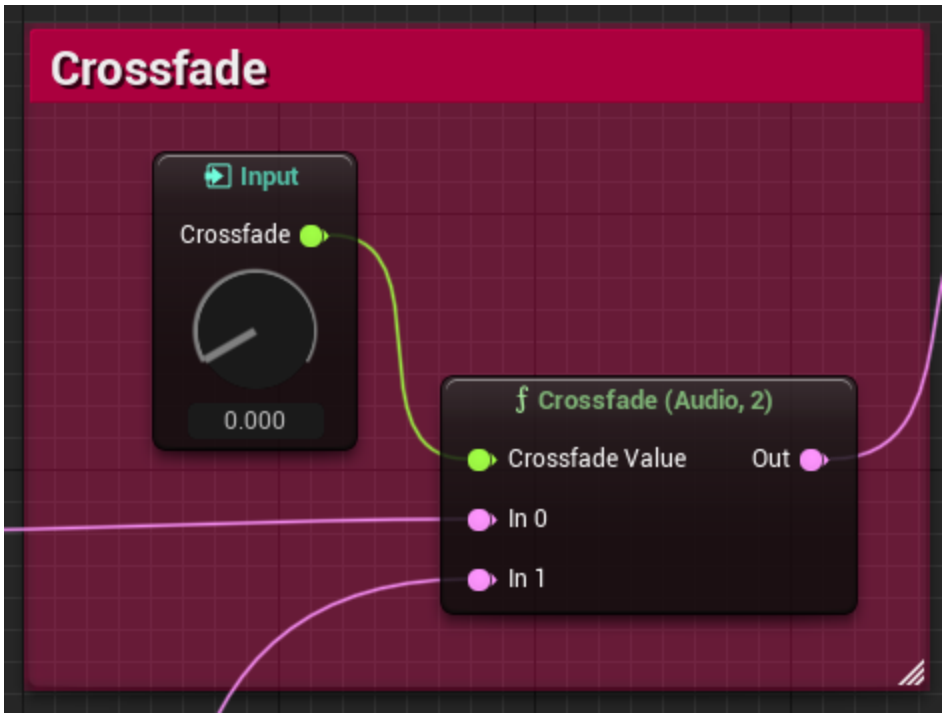


Build the Synthesis (Saw) section to produce a saw wave at the frequency of the input note.

1. Select all of the nodes in the Synthesis (Sine) section, right-click one of the selected nodes, and select **Duplicate**.
2. Ensure that the input pins of the **Add (Float)** and **MIDI to Frequency (Float)** nodes are connected to the output pin of the **Add (Float)** node from the Melody Generation section.
3. Delete both of the **Sine** nodes and replace them with **Saw** nodes.

4. Enclose the new nodes in a comment box named "Synthesis (Saw)."

1.6 - Crossfade



Build the Crossfade section to control the ratio of the sine and saw synths.

1. On the **Add (Audio)** in the Synthesis (Sine) section, drag off the output pin and create a **Crossfade (Audio, 2)** node.
2. On the **Crossfade (Audio, 2)** node:
    a. Connect the **In 1** pin to the output pin on the **Add (Audio)** node in the Synthesis (Saw) section.
    b. Drag off the **Crossfade Value** pin and select **Promote to Graph Input**. This creates a **Float Input** node named Crossfade Value.
3. Select Crossfade Value.
4. In the **Details** panel, set **General > Input** to "Crossfade" to rename the input.
5. Enclose the new nodes in a comment box named "Synthesis (Crossfade)."

1.7 - Filtering

Build the Filtering section to smooth out the sound.

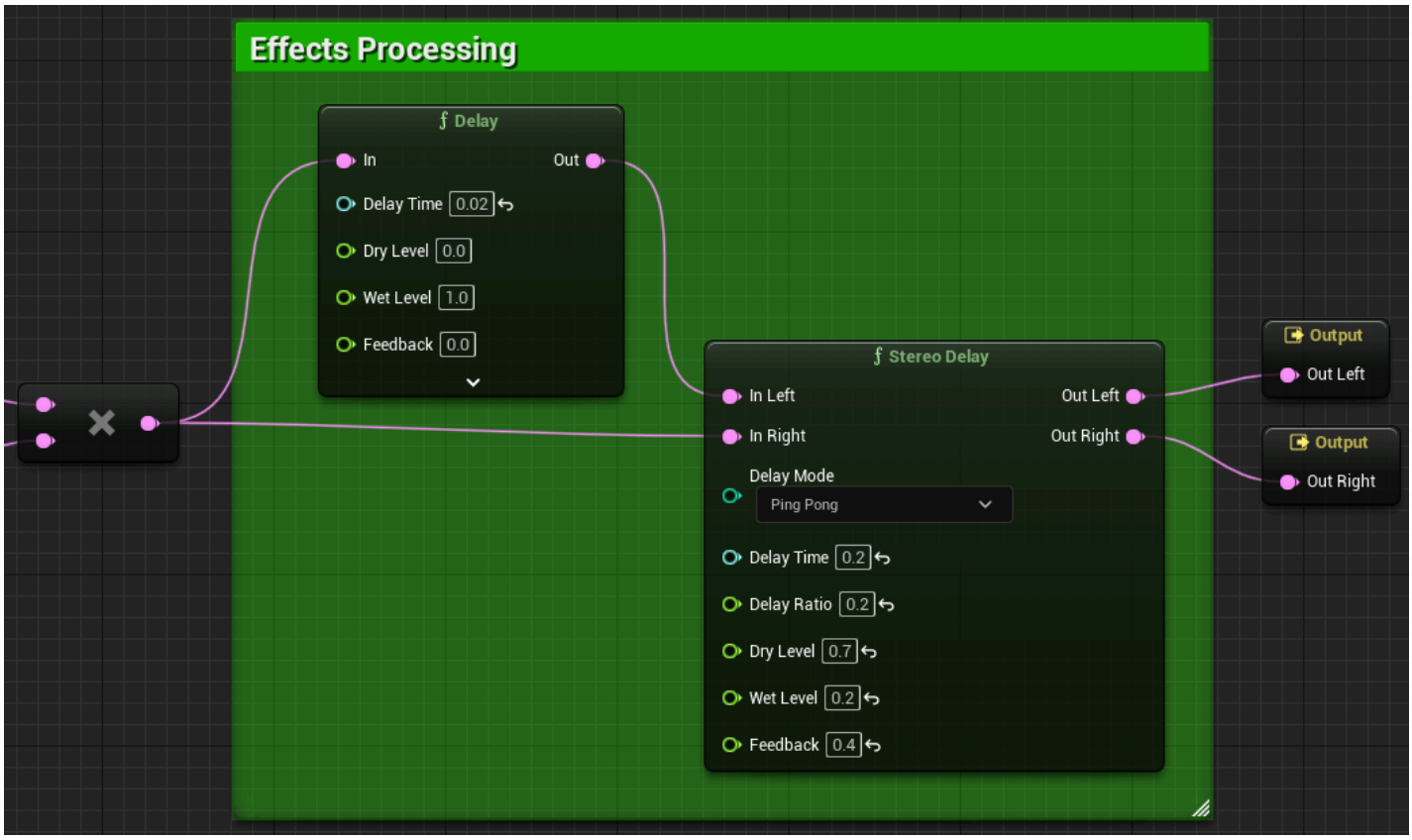
1. On the **Crossfade (Audio, 2)** node in the Crossfade section, drag off the **Out** pin and create a **Ladder Filter** node.
2. On the **Ladder Filter** node:
    a. Set **Resonance** to 6.0.
    b. Drag off the **Cutoff Frequency** pin and create an **LFO** node.
3. On the **LFO** node:
    a. Set **Frequency** to 0.5.
    b. Set **Min Value** to 500.0.
    c. Set **Max Value** to 5000.0.
4. Enclose the new nodes in a comment box named "Filtering."

1.8 - Enveloping

Build the Enveloping section with an attack-decay envelope to remove the sustain of the notes in the melody.

1. On the **Random Get (Float:Array)** node in the Melody Generation section, drag off the **On Next** pin and create an **AD Envelope (Audio)** node.
2. On the **AD Envelope (Audio)** node, set **Decay Time** to 0.1.
3. Enclose the new node in a comment box named "Enveloping."

1.9 - Effects Processing



Build the Effects Processing section to produce a stereo widening effect. The delay creates an illusion that the mono signal is a stereo signal.

1. On the **AD Envelope (Audio)** node in the Enveloping section, drag off the **Out Envelope** pin and create a **Multiply (Audio)** node.
2. On the **Multiply (Audio)** node:
    a. Connect the bottom multiplicand pin to the **Out** pin of the **Ladder Filter** node in the Filtering section.
    b. Drag off the output pin and create a **Delay** node.
3. On the **Delay** node:
    a. Set **Delay Time** to 0.02.
    b. Drag off the **Out** pin and create a **Stereo Delay** node.
4. On the **Stereo Delay** node:
    a. Connect the **In Right** pin to the output pin of the **Multiply (Audio)** node.
    b. Set **Delay Mode** to Ping Pong.
    c. Set **Delay Time** to 0.2.
    d. Set **Delay Ratio** to 0.2.
    e. Set **Dry Level** to 0.7.
    f. Set **Wet Level** to 0.2.
    g. Set **Feedback** to 0.4.

h. Connect the **Out Left** pin to the **Out Left Output** node.

i. Connect the **Out Right** pin to the **Out Right Output** node.

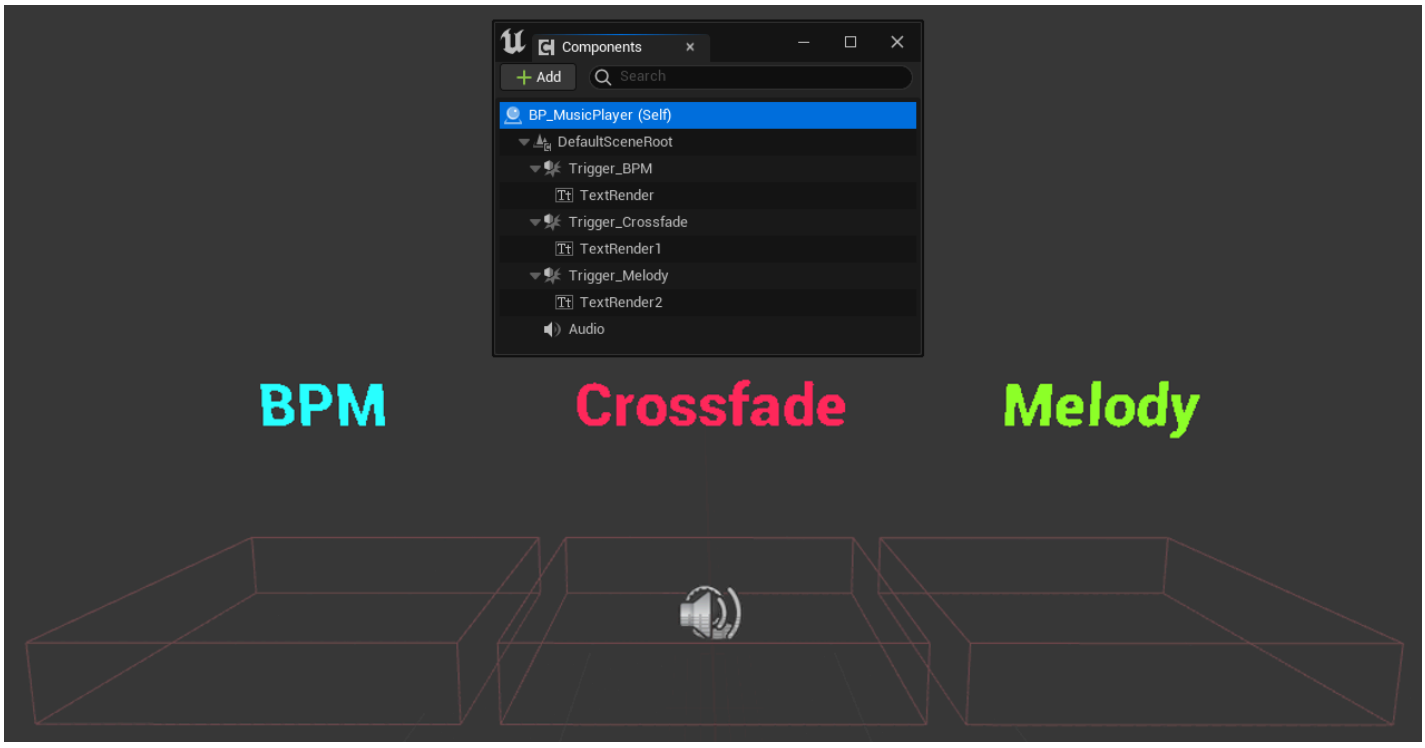
5. Enclose the **Delay** and **Stereo Delay** nodes in a comment box named "Effects Processing."

1.11 - Play the MetaSound

The MetaSound is now ready to play.

1. Save the MetaSound.
2. Click the **Play** button on the **MetaSound Editor Toolbar** to play the MetaSound.
3. Use the widgets or the **Details** panel to adjust input values and listen to the changes in real-time.

2 - Build the Blueprint Actor



Create a **Blueprint Actor** with **Box Colliders** that triggers changes in the MetaSound during runtime.

2.1 - Create the Blueprint Class

1. In the **Content Browser**, click the **Add** button.
2. Select **Blueprint Class**.
3. From the **Pick Parent Class** window, select **Actor**.
4. Give the newly created Blueprint Actor a name, such as `BP_MusicPlayer`.

2.2 - Add Components

Add Components to the Actor to create three separate trigger zones that respond when the Player moves in and out of them.

1. Double-click the Blueprint Actor to open the **Blueprint Editor**.
2. Add a Box Collision Component.
    a. In the **Components** panel:
        i. Click the **Add** button.
        ii. Type "Box Collision" into the search bar and press Enter.
    b. In the **Details** panel:
        i. Set **Transform > Scale** to 5.0, 5.0, 1.0.
        ii. Disable **Rendering > Hidden In Game**.
3. Add a Text Render Component to the Box Collision Component.
    a. In the **Components** panel:
        i. Select the Box Collision Component.
        ii. Click the **Add** button.
        iii. Type "Text Render" into the search bar and press Enter.
    b. In the **Details** panel:
        i. Set **Transform > Location** to 0.0, 0.0, 150.0.
        ii. Set **Text > Horizontal Alignment** to Center.
        iii. Set **Text > World Size** to 64.0.
4. Create two additional copies of the Components. Do the following twice:
    a. In the **Components** panel, Ctrl + click to select both the Box Collision Component and Text Render Component.
    b. Right-click and select **Duplicate**.
5. Customize the BPM Trigger.
    a. In the **Components** panel, right-click the first Box Collision Component, select **Rename**, and name it `Trigger_BPM`.

    b. In the **Details** panel, set **Transform > Location** to 0.0, 350.0, 0.0.
    c. In the **Components** panel, select the child Text Render Component.
    d. In the **Details** panel, set **Text > Text** to BPM.
6. Customize the Crossfade Trigger.
    a. In the **Components** panel, right-click the second Box Collision Component, select **Rename**, and name it `Trigger_Crossfade`.
    b. Select the child Text Render Component.
    c. In the **Details** panel, set **Text > Text** to Crossfade.
7. Customize the Melody Trigger.
    a. In the **Components** panel, right-click the third Box Collision Component, select **Rename**, and name it `Trigger_Melody`.
    b. In the **Details** panel, set **Transform > Location** to 0.0, -350.0, 0.0.
    c. In the **Components** panel, select the child Text Render Component.
    d. In the **Details** panel, set **Text > Text** to Melody.
8. Add an Audio Component for the MetaSound.
    a. In the **Components** panel:
        i. Click the **Add** button.
        ii. Type "Audio" into the search bar and press Enter.
        iii. In the **Details** panel, set **Sound > Sound** to the Music MetaSound.

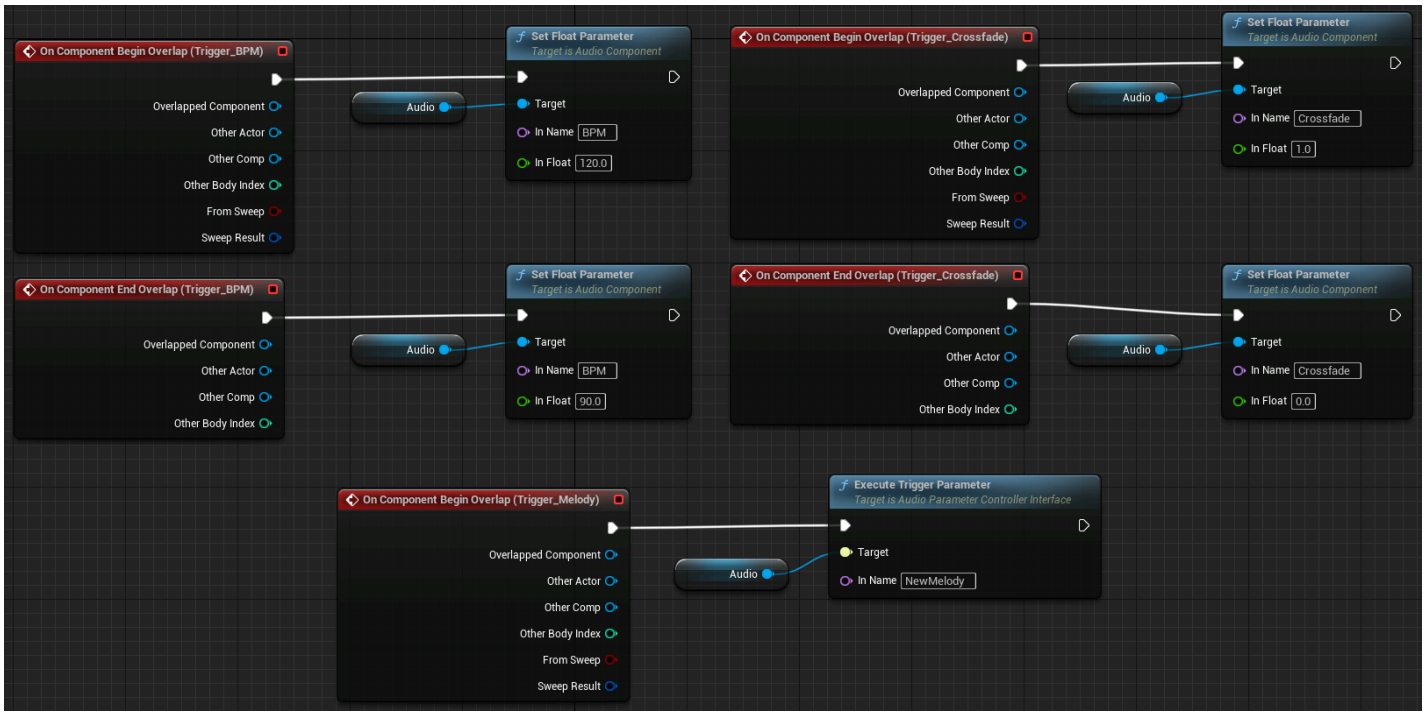
2.3 - Add Trigger Events

Add collision response events for each of the zones.

1. In the **Components** panel, select Trigger_BPM.
2. In the **Details** panel:
    a. Select the **Add (+)** button next to **Events > On Component Begin Overlap**.
    b. Select the **Add (+)** button next to **Events > On Component End Overlap**.
3. In the **Components** panel, select Trigger_Crossfade.
4. In the **Details** panel:
    a. Select the **Add (+)** button next to **Events > On Component Begin Overlap**.
    b. Select the **Add (+)** button next to **Events > On Component End Overlap**.
5. In the **Components** panel, select Trigger_Melody.

6. In the **Details** panel, select the **Add (+)** button next to **Events > On Component Begin Overlap**.

2.4 - Build Trigger Events



Click image for full size.

Attach nodes to each event to control the music.

> You might need to turn off the **Context Sensitive** filter located at the top-right of the search context menu to find the **Set Float Parameter (Audio)** and **Execute Trigger Parameter** nodes used below.

1. On the **On Component Begin Overlap (Trigger_BPM)** node, drag off the **Exec Output (>)** pin, and create a **Set Float Parameter (Audio)** node.
2. On the **Set Float Parameter** node:
    a. Set **In Name** to BPM.
    b. Set **In Float** to 120.0.
3. On the **On Component End Overlap (Trigger_BPM)** node, drag off the **Exec Output (>)** pin, and create another **Set Float Parameter (Audio)** node.
4. On the new **Set Float Parameter** node:

    a. Set **In Name** to BPM.
    b. Set **In Float** to 90.0.
5. On the **On Component Begin Overlap (Trigger_Crossfade)** node, drag off the **Exec Output (>)** pin, and create another **Set Float Parameter (Audio)** node.
6. On the new **Set Float Parameter** node:
    a. Set **In Name** to Crossfade.
    b. Set **In Float** to 1.0.
7. On the **On Component End Overlap (Trigger_Crossfade)** node, drag off the **Exec Output (>)** pin, and create another **Set Float Parameter (Audio)** node.
8. On the new **Set Float Parameter** node:
    a. Set **In Name** to Crossfade.
    b. Set **In Float** to 0.0.
9. On the **On Component Begin Overlap (Trigger_Melody)** node, drag off the **Exec Output (>)** pin, and create an **Execute Trigger Parameter** node.
10. On the **Execute Trigger Parameter** node:
    a. Set **In Name** to NewMelody.
    b. In the **Components** panel, drag the Audio Component onto the **Target** pin.
11. Compile and save the Blueprint.

3 - Test the Level

The Blueprint is ready to test.

1. Place the BP_MusicPlayer by dragging it from the **Content Browser** into the **Level**.
2. Using the transformation widget, place the Actor so your character can move in and out of it.
3. Click the **Play** button on the **Level Editor Toolbar**.
4. Move in and out of the triggers and observe their effects on the music.