

Developer

/ Documentation

/ Unreal Engine ▾

/ Unreal Engine 5.4 Documentation

/ Working with Audio

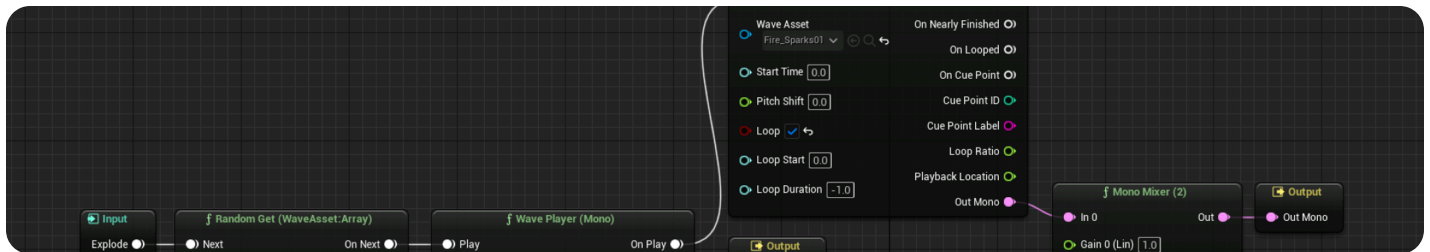
/ Sound Sources

/ MetaSounds

/ MetaSounds Quick Start

MetaSounds Quick Start

A quick guide on getting started with MetaSounds.



⚠ Learn to use this **Beta** feature, but use caution when shipping with it.

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

In this guide, you will learn how to create two gameplay-driven **MetaSound Sources**: a bomb sound effect and ambient wind.

Prerequisites

Before creating your first MetaSounds with this guide, you must create a new [First Person Template](#) project with **Starter Content**.

1 - Create the Bomb MetaSound

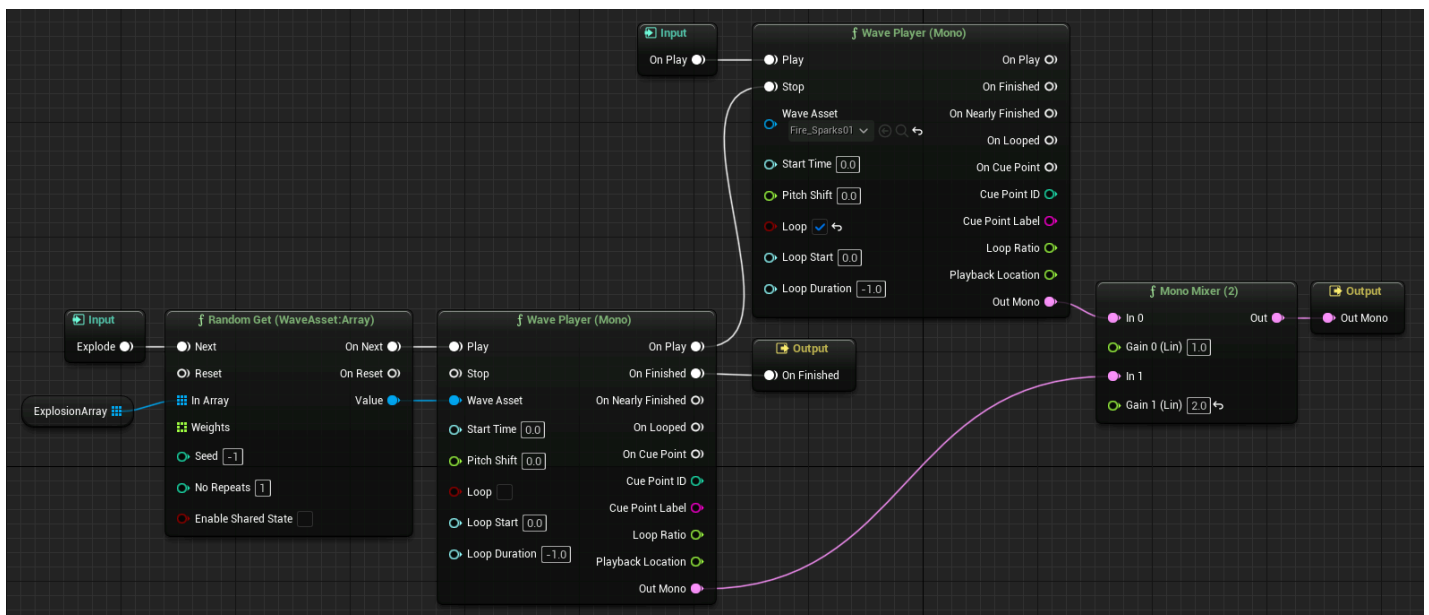
Though the rifle plays a sound when fired, the projectile is silent. Using MetaSound, you will create a bomb sound effect to be used by the projectile.

1A - Create the MetaSound Source

Because this will be a sound source within 3D space, start by creating a MetaSound Source with an attached **Sound Attenuation Asset**.

1. Create a MetaSound Source.
 - a. In the **Content Browser**, click the **Add** button.
 - b. Select **Audio > MetaSound Source**.
 - c. Give the newly created Asset a name (such as MSS_Bomb).
2. Double-click the MetaSound to open the **MetaSound Editor**.
3. Set the **Attenuation Settings** to spatialize and attenuate the MetaSound based on its position relative to the listener.
 - a. Click the **Source** button on the **MetaSound Editor Toolbar**.
 - b. In the **Details** panel, click the dropdown next to **Attenuation > Attenuation Settings**.
 - c. Select **Sound Attenuation** under the Create New Asset heading.
 - d. Give the newly created Asset a name (such as SA_Bomb), then save it.

1B - Build the MetaSound Graph

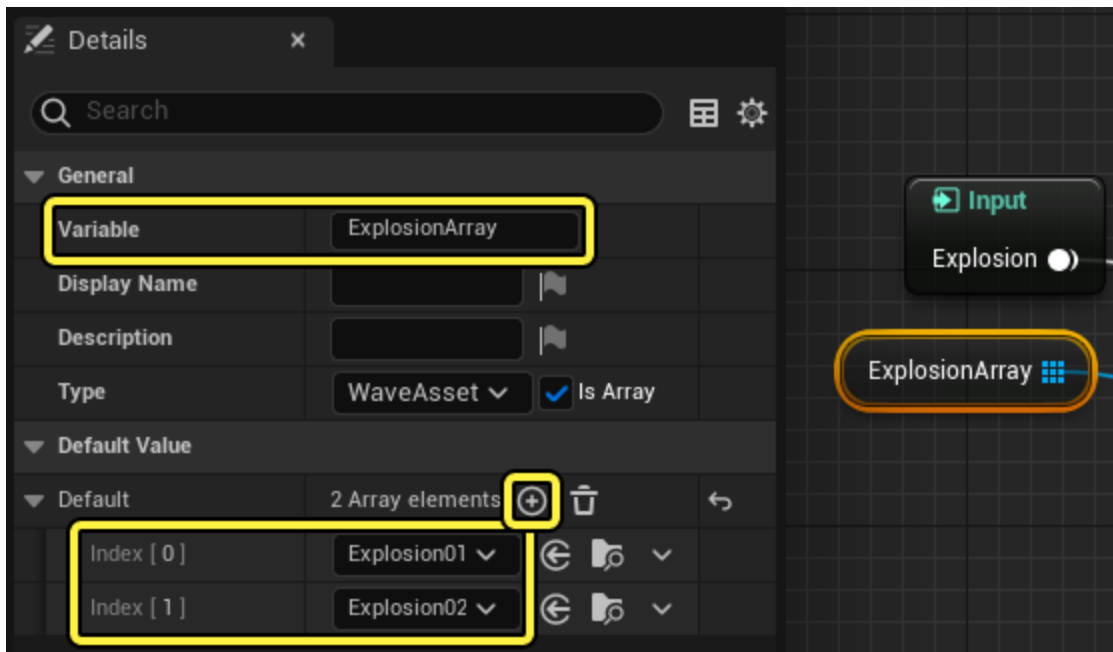


Click image for full size.

Build the **MetaSound Graph** that controls how your MetaSound Source sounds. Follow the instructions below to create the graph shown above.

1. Find the **On Play Input** node and drag off of the pin into an empty space. Enter "Wave Player (Mono)" into the node search to create a connected node. You can move the node around the graph by dragging it.
2. On the **Wave Player (Mono)** node:
 - a. Click the **Wave Asset** dropdown and select the Fire_Sparks01 Sound Wave.
 - b. Enable **Loop**.
 - c. Drag off the **Out Mono** pin and create a **Mono Mixer (2)**.
 - d. Drag off the **Stop** pin and create another **Wave Player (Mono)**.
3. On the new **Wave Player (Mono)** node:
 - a. Connect the **On Finished** pin to the **On Finished Output** node.
 - b. Connect the **Out Mono** pin to the **In 1** pin on the **Mono Mixer (2)**.
 - c. Drag off the **Play** pin and create a **Random Get (WaveAsset:Array)** node.
 - d. Connect the **Wave Asset** pin to the **Value** pin on the **Random Get (WaveAsset:Array)** node.
4. On the **Random Get (WaveAsset:Array)** node:
 - a. Drag off the **Next** pin and select **Promote to Graph Input**. This will create a **Trigger Input** node named Next.
 - b. Drag off the **In Array** pin and select **Promote to Graph Variable**. This will create a **WaveAsset:Array Variable** node named In Array.
5. On the **Mono Mixer (2)**:
 - a. Enter 2.0 for **Gain 1 (Lin)**.
 - b. Connect the **Out** pin to the **Out Mono Output** node.

1C - Adjust the Explosion Wave Player Inputs



Adjust the Next Trigger and the In Array Variable.

1. Select the Next **Trigger Input** node.
2. In the **Details** panel, enter "Explode" into **General > Input** to rename it. This will be the name of the Trigger you will set to execute from Blueprint.
3. Select the In Array **WaveAsset:Array Variable** node.
4. In the **Details** panel, enter "ExplosionArray" into **General > Variable** to rename it.
5. Click the **Add (+)** button twice for **Default Value > Default**. These indices will hold Sound Wave references to be randomly selected from.
6. Click the **Index[0]** dropdown and select the Explosion01 Sound Wave.
7. Click the **Index[1]** dropdown and select the Explosion02 Sound Wave.

1D - Play the MetaSound



The MetaSound is now ready to play.

1. Click the **Play** button on the **MetaSound Editor Toolbar** to play the MetaSound. A spark sound will loop until the **Explode** Trigger executes, at which point a random explosion sound in the **ExplosionArray** will play before the MetaSound finishes. You can simulate

the **Explode** Trigger execution by clicking the **Execute (Down Arrow)** button on the top-right corner of the **Trigger Input** node while the MetaSound is playing.

2. Save the MetaSound Source and close the MetaSound Editor.

2 - Add Bomb Logic to Blueprint

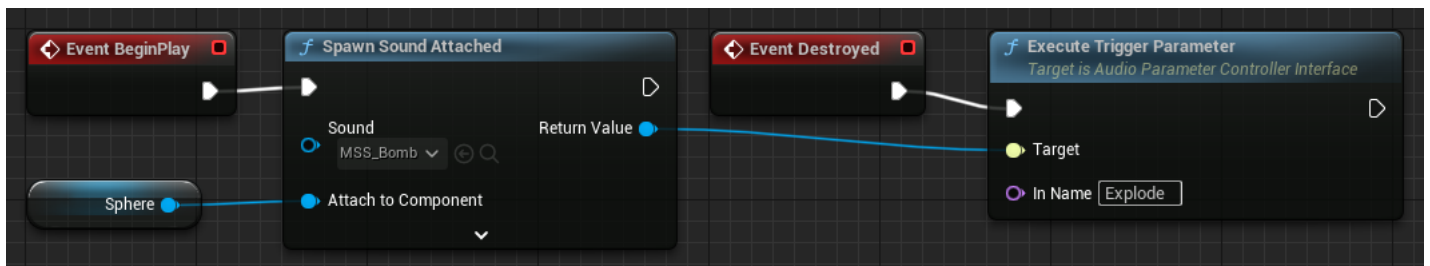
After designing the bomb sound, use the projectile's Blueprint to set up the runtime logic.

2A - Open the Projectile Blueprint

Open the pre-built projectile Blueprint.

1. In the **Content Browser**, navigate to `All/Content/FirstPerson/Blueprints`.
2. Double-click **BP_FirstPersonProjectile** to open it in the **Blueprint Editor**.

2B - Build the Blueprint Graph



Add to the existing **Blueprint Graph** to control the bomb MetaSound based on projectile lifecycle.

1. Right-click in an empty space and create an **Event BeginPlay** node.
2. Drag off the **Event BeginPlay** node and create a **Spawn Sound Attached** node.
3. On the **Spawn Sound Attached** node:
 - a. Click the **Sound** dropdown and select your bomb MetaSound.
 - b. Drag off the **Attach to Component** pin and create a **Get Sphere** node.
 - c. Drag off the **Return Value** pin and create an **Execute Trigger Parameter** node.
 - d. Hold Alt and click the **Exec Output (>)** pin to remove the connection between the **Spawn Sound Attached** and **Execute Trigger Parameter** nodes.
4. On the **Execute Trigger Parameter** node:
 - a. Enter "Explode" for **In Name**.

- b. Drag off the **Exec Input (>)** pin and create an **Event Destroyed** node.
5. Save your changes to the Blueprint and close the Blueprint Editor.

2C - Test the Level

The Blueprint is now ready to test.

Click the **Play** button on the **Level Editor Toolbar**, pick up the rifle (by moving into it), and shoot it (by left-clicking) to verify your work.

The projectiles will produce spatialized, dynamic sounds. The spark sound loops until the projectile explodes by either making contact with a blue box or after a short time passes.

3 - Create the Wind MetaSound

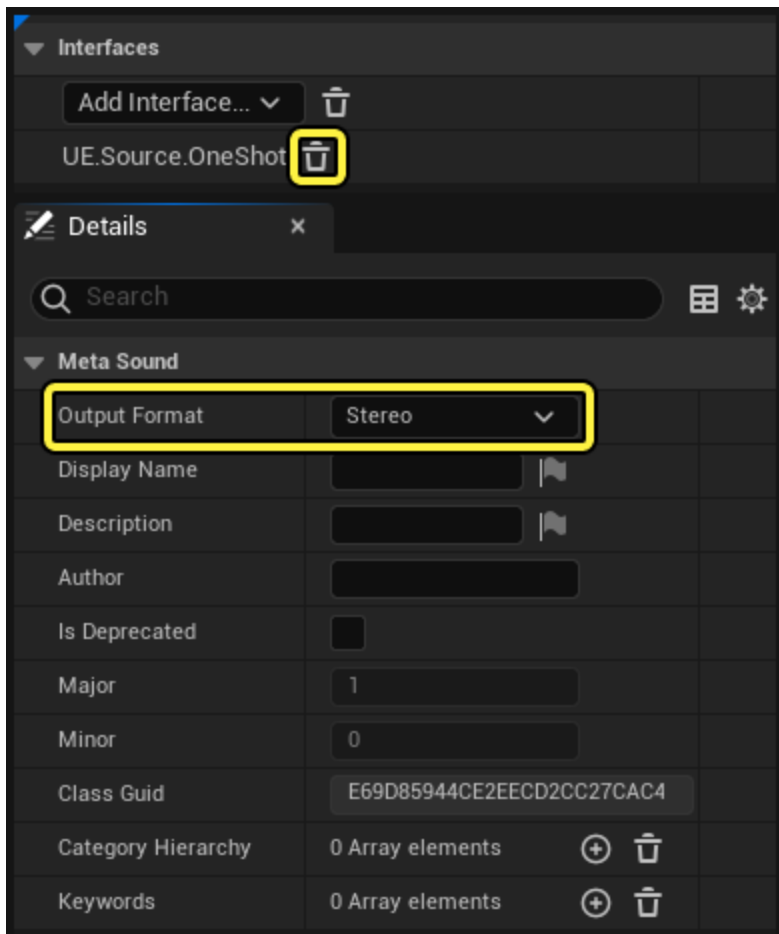
Create a MetaSound for ambient wind to add some atmosphere to the scene.

3A- Create the MetaSound Source

Create another MetaSound Source Asset. This sound will not be spatialized within 3D space, so you don't have to attach a Sound Attenuation Asset.

1. Create a MetaSound Source.
 - a. In the **Content Browser**, click the **Add** button.
 - b. Select **Audio > MetaSound Source**.
 - c. Give it a name (such as MSS_Wind).
2. Double-click the MetaSound to open the MetaSound Editor.

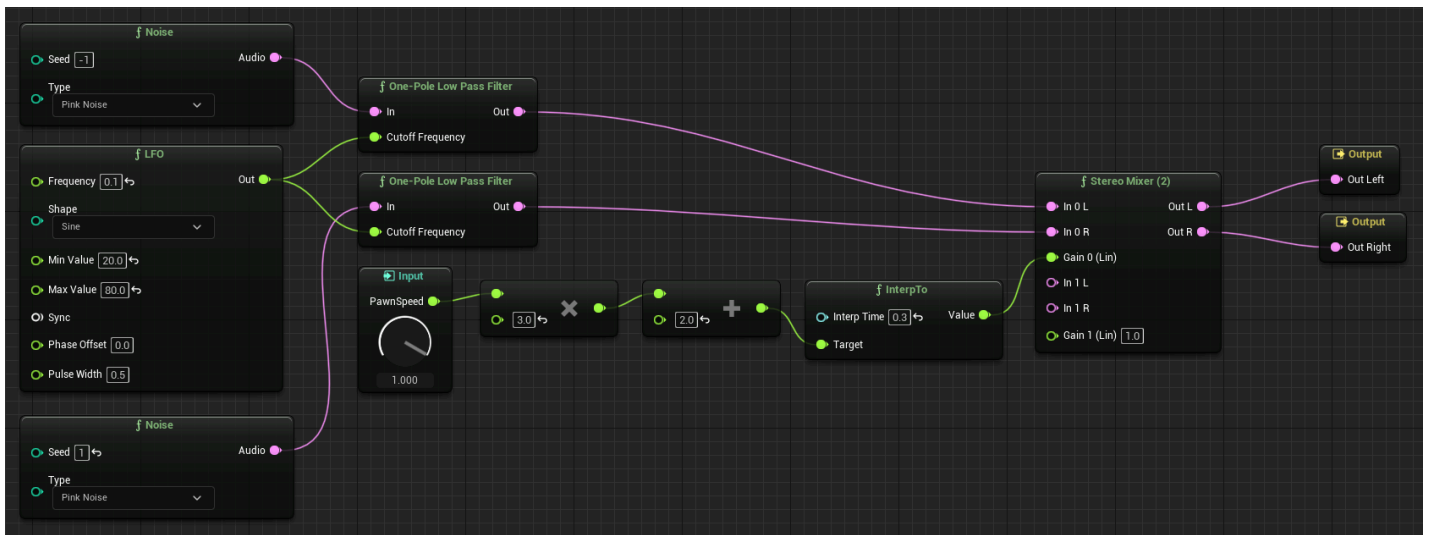
3B - Adjust the MetaSound Details



Before building the graph, adjust the default properties of the MetaSound to support persistent stereo audio.

1. In the Interfaces panel, click the **Remove (Trash Bin)** button next to the **UE.Source.OneShot** Interface entry. This will remove the **On Finished Output** node, which isn't used on persistent sounds such as ambience or music.
2. Click the **MetaSound** button on the **MetaSound Editor Toolbar**.
3. In the **Details** panel, click the **MetaSound > Output Format** dropdown and select **Stereo**. This will replace the **Out Mono Output** node with **Out Left** and **Out Right Output** nodes.

3C - Build the MetaSound Graph



Click image for full size.

Build the **MetaSound Graph** that controls how your MetaSound Source sounds. Follow the instructions below to create the graph shown above.

1. Select the **On Play Input** node and delete it using the Delete key. It will not be needed in this graph.
2. Drag off the **Out Left** pin and create a **Stereo Mixer (2)** node.
3. Connect the **Out Right** pin to the **Out R** pin on the **Stereo Mixer (2)** node.
4. Right-click in an empty space and create a **Noise** node.
5. On the **Noise** node, drag off the **Audio** pin and create a **One-Pole Low Pass Filter** node.
6. Drag a selection box to select the **Noise** and **One-Pole Low Pass Filter** nodes.
7. Right-click either selected node and select **Duplicate**.
8. On the new **Noise** node, enter 1 for **Seed**. This will introduce variance in the noise generation.
9. On one of the **One-Pole Low Pass Filter** nodes, connect the **Out** pin to the **In 0 L** pin on the **Stereo Mixer (2)** node.
10. On the other **One-Pole Low Pass Filter** node, connect the **Out** pin to the **In 0 R** pin on the **Stereo Mixer (2)** node.
11. Right-click in an empty space and create an **LFO** node.
12. On the **LFO** node:
 - a. Enter 0.1 for **Frequency**.
 - b. Enter 20.0 for **Min Value**.
 - c. Enter 80.0 for **Max Value**.
 - d. Connect the **Out** pin to both of the **Cutoff Frequency** pins on the **One-Pole Low Pass Filter** nodes.

13. On the **Stereo Mixer (2)** node, drag off the **Gain 0 (Lin)** pin and create an **InterpTo** node.
14. On the **InterpTo** node:
 - a. Enter 0.3 for **Interp Time**.
 - b. Drag off the **Target** pin to create an **Add (Float)** node.
15. On the **Add (Float)** node:
 - a. Enter 2.0 for **Bottom Addend**.
 - b. Drag off the **Top Addend** pin and create a **Multiply (Float)** node.
16. On the **Multiply (Float)** node:
 - a. Enter 3.0 for **Bottom Multiplicand**.
 - b. Drag off the **Top Multiplicand** pin and select **Promote to Graph Input**. This will create a **Float Input** node named PrimaryOperand.
17. Select the PrimaryOperand **Float Input** node.
18. In the **Details** panel, enter "PawnSpeed" into **General > Input** to rename it.

3D - Play the MetaSound



The MetaSound is now ready to play.

1. Click the **Play** button on **MetaSound Editor Toolbar** to play the MetaSound. A dynamic wind-like sound will play in stereo until you click the **Stop** button. The signal gain is affected by the PawnSpeed **Float Input** node. You can simulate PawnSpeed values by clicking the **Input Widget (Dial)** on the node and dragging up or down.
2. Save the MetaSound Source and close the MetaSound Editor.

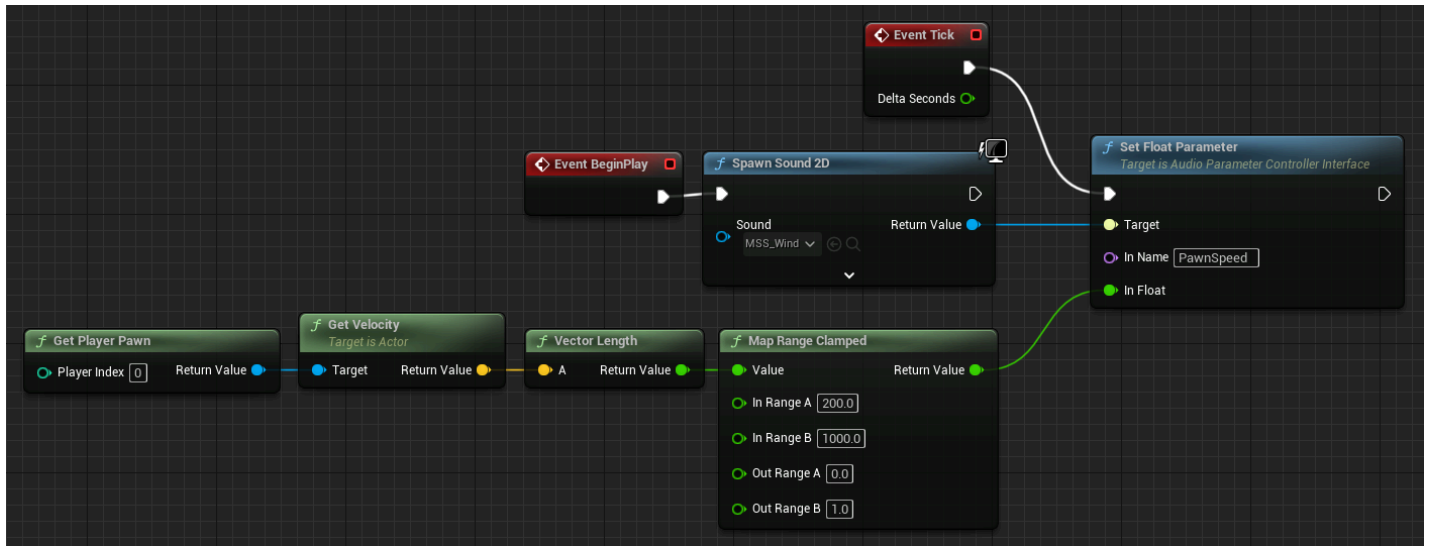
4 - Add Wind Logic to Blueprint

After designing the wind sound, use the **Level Blueprint** to set up the runtime logic.

4A - Open the Level Blueprint

To open the Level Blueprint, click the **Blueprint** button in the **Level Editor Toolbar** and select **Open Level Blueprint**.

4B - Build the Blueprint Graph



Click image for full size.

Build the Blueprint Graph to control the wind MetaSound based on Player movement.

1. Right-click in an empty space and create an **Event BeginPlay** node.
2. Drag off the **Event BeginPlay** node and create a **Spawn Sound 2D** node.
3. On the **Spawn Sound 2D** node:
 - a. Click the **Sound** dropdown and select your wind MetaSound.
 - b. Drag off the **Return Value** and create a **Set Float Parameter** node.
 - c. Hold Alt and click the **Exec Output (>)** pin to remove the connection between the **Spawn Sound 2D** and **Set Float Parameter** nodes.
4. On the **Set Float Parameter** node:
 - a. Enter "PawnSpeed" for **In Name**.
 - b. Drag off the **Exec Input (>)** pin and create an Event Tick node.
5. Right-click in an empty space and create a **Get Player Pawn** node.

6. On the **Get Player Pawn** node, drag off the **Return Value** pin and create a **Get Velocity** node.
7. On the **Get Velocity** node, drag off the **Return Value** pin and create a **Vector Length** node.
8. On the **Vector Length** node, drag off the **Return Value** pin and create a **Map Range Clamped** node.
9. On the **Map Range Clamped** node:
 - a. Enter 200.0 for **In Range A**.
 - b. Enter 1000.0 for **In Range B**.
 - c. Enter 1.0 for **Out Range B**.
 - d. Connect the **Return Value** pin to the **In Float** pin on the **Set Float Parameter** node.
10. Save your changes to the Blueprint and close the Blueprint Editor.

4C - Test the Level

The Blueprint is ready to test.

Click the **Play** button on the **Level Editor Toolbar** and move around to verify your work.

The wind sound will play at a low volume when idle and grow louder as your velocity increases.

5 - On Your Own!

Now that you've finished creating two basic MetaSounds, consider taking this project even further. Below are some suggestions you can explore on your own:

- Alter your MetaSounds with some additional nodes. See the [MetaSounds Reference Guide](#) for information on the node library and other MetaSound features.
- Replace the default rifle firing sound with a MetaSound. Consider altering the sound dynamically based on the rate of fire, shot angle, or other variables.
- Add MetaSounds for player actions such as walking, jumping, and item pickup.
- Improve on the simple wind implementation you created with this guide. See the [Lyra Sample Game](#) project for an example of an advanced implementation that supports additional features, such as panning and environment response.