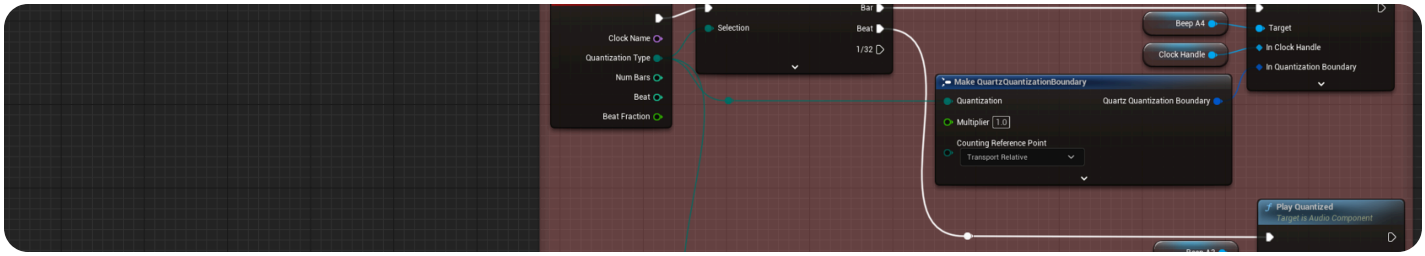
Quartz Quick Start

A quick guide on getting started with Quartz.



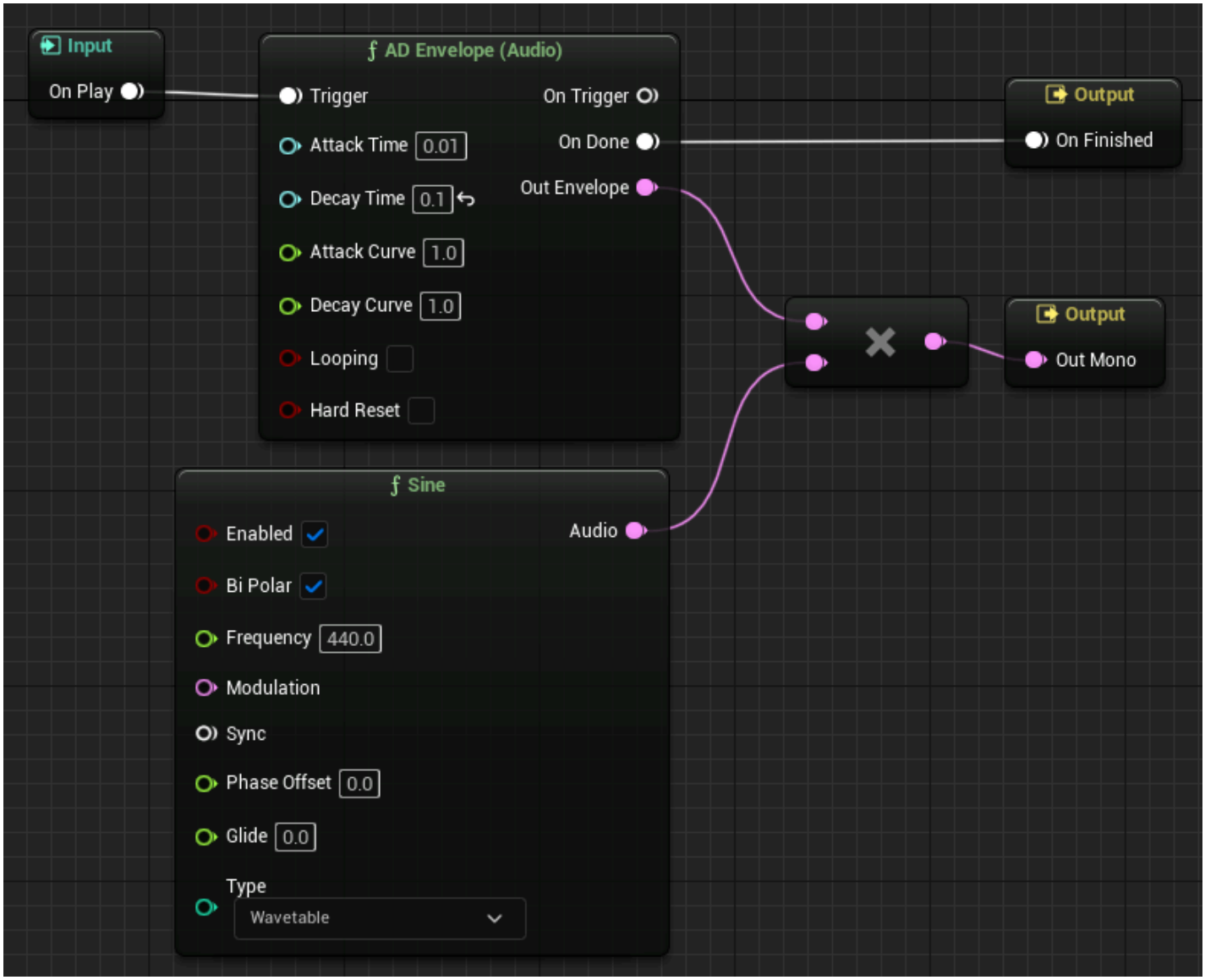
Quartz is a Blueprint-exposed scheduling system that solves timing issues between the game, audio logic, and audio rendering threads to provide sample-accurate audio playback.

This guide teaches you how to create a Quartz-powered metronome that triggers audio and gameplay events.

Prerequisites

Create a new Third Person Template project.

1 - Create the MetaSounds



Create two **MetaSound Sources** for the metronome beeps. Follow the steps below to build the graph above.

1.1 - Create the High-Frequency Beep

1. Create a MetaSound Source.
    a. In the **Content Browser**, click the **Add** button.
    b. Select **Audio > MetaSound Source**.
    c. Name the new MetaSound `MSS_BeepA4`.
2. Double-click the MetaSound to open the **MetaSound Editor**.
3. Find the **On Play Input** node and drag off of the pin into an empty space. Enter "AD Envelope (Audio)" into the node search to create a connected node. You can move the node around the graph by dragging it.

4. On the **AD Envelope (Audio)** node:
    a. Set **Decay Time** to 0.1.
    b. Connect the **On Done** pin to the **On Finished Output** node.
    c. Drag off the **Out Envelope** pin and create a **Multiply (Audio)** node.
5. On the **Multiply (Audio)** node:
    a. Drag off the bottom multiplicand pin and create a **Sine** node.
    b. Connect the output pin to the **Out Mono Output** node.
6. Click the **Play** button on the **MetaSound Editor Toolbar** to play the short, high-frequency beep sound.
7. Save the MetaSound and close the **MetaSound Editor**.

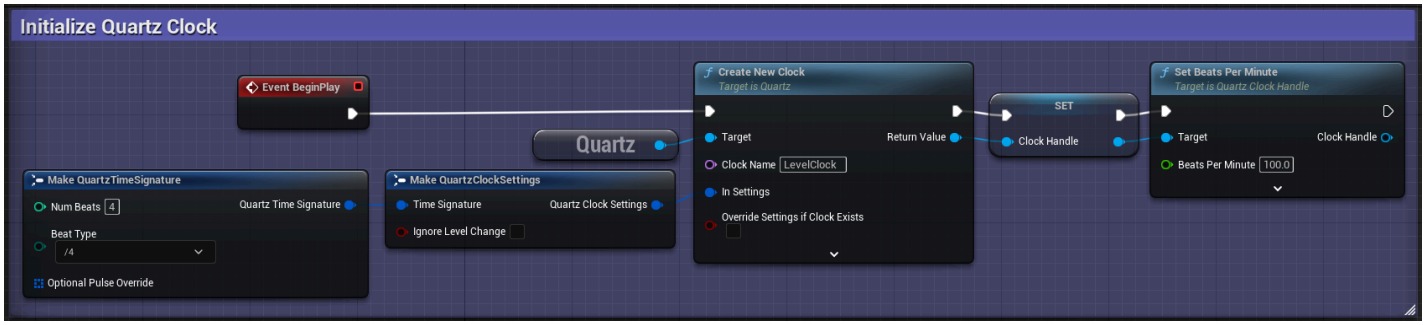
1.2 - Create the Low-Frequency Beep

1. In the **Content Browser**, right-click the `MSS_BeepA4` MetaSound and select **Duplicate**.
2. Name the new MetaSound `MSS_BeepA3`.
3. Double-click the MetaSound to open the **MetaSound Editor**.
4. On the **Sine** node, set the **Frequency** to 220.
5. Click the **Play** button on the **MetaSound Editor Toolbar** to play the short, low-frequency beep sound.
6. Save the MetaSound and close the **MetaSound Editor**.

2 - Build the Level Blueprint

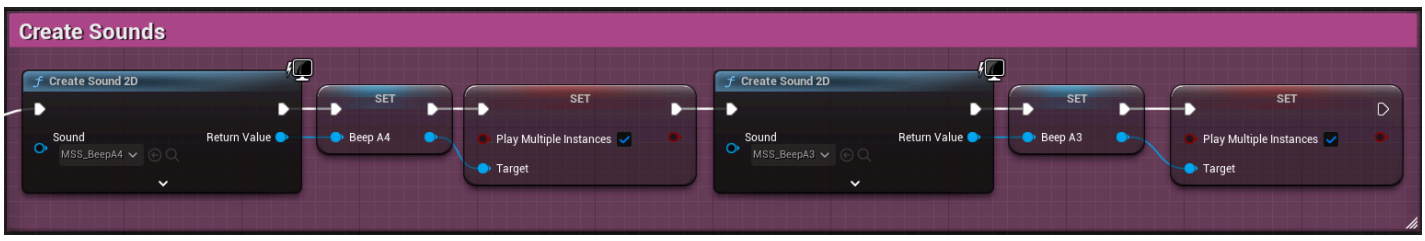
Construct the Level Blueprint to create a Quartz Clock, sounds, and the event delegate for the metronome.

2.1 - Create the Quartz Clock



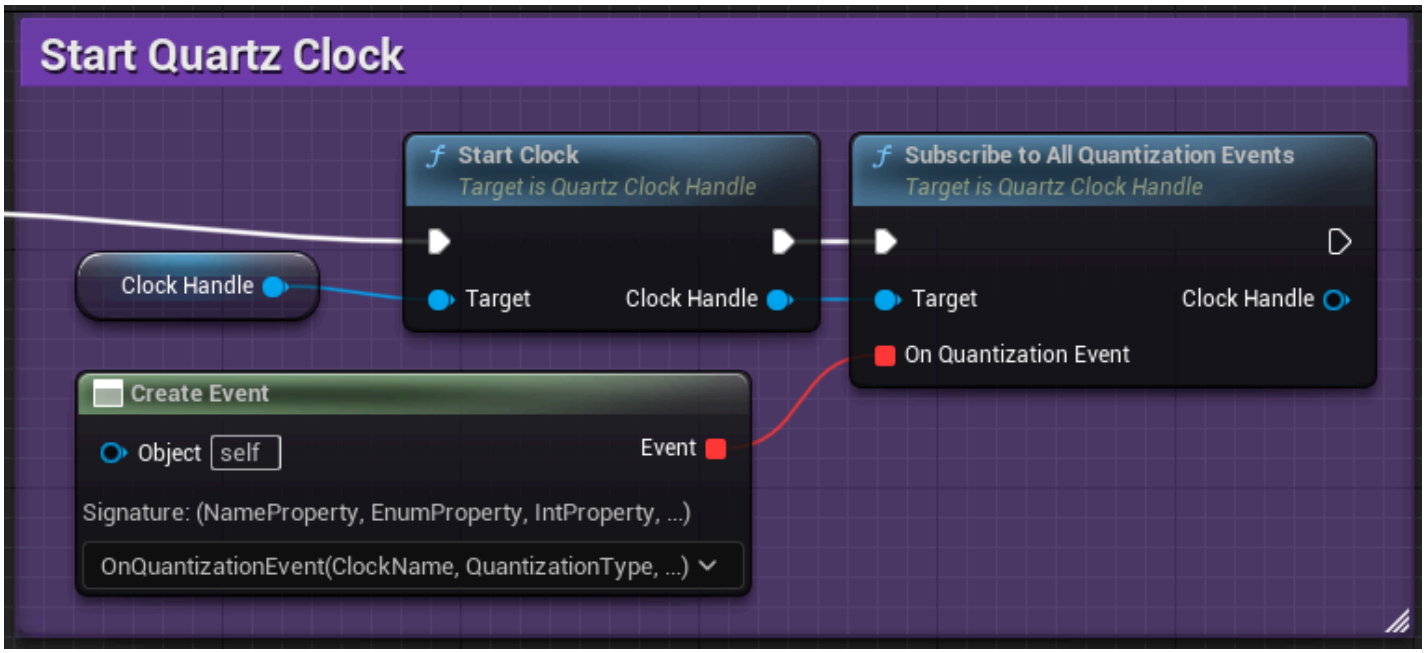
1. On the **Level Editor Toolbar**, click the **Blueprint** button and select **Open Level Blueprint**.
2. Right-click in an empty space and add a **Get QuartzSubsystem** node.
3. On the **Get QuartzSubsystem** node, drag off the output pin and add a **Create New Clock** node.
4. On the **Create New Clock** node:
    a. Connect the **Exec Input (>)** pin to the **Event BeginPlay** node.
    b. Set **Clock Name** to `LevelClock`.
    c. Drag off the **In Settings** pin and add a **Make QuartzClockSettings** pin.
    d. Drag off the **Return Value** pin and select **Promote to Variable**. This creates a **Set** node and a Blueprint variable named `NewVar` to store the Quartz Clock Handle and prevent its' garbage collection.
    e. Connect the **Exec Output (>)** pin to the **Set (NewVar)** node.
5. In the **My Blueprint** panel, right-click the **NewVar** variable and select **Rename**.
6. Name the variable `ClockHandle`.
7. On the **Make QuartzClockSettings** node, drag off **Time Signature** and create a **Make QuartzTimeSignature** node.
8. On the **Set (Clock Handle)** node:
    a. Drag off the output pin and create a **Set Beats Per Minute** node.
    b. Connect the **Exec Output (>)** pin to the **Set Beats Per Minute** node.
9. On the **Set Beats Per Minute** node, set **Beats Per Minute** to 100.0.

2.2 - Create the Sounds



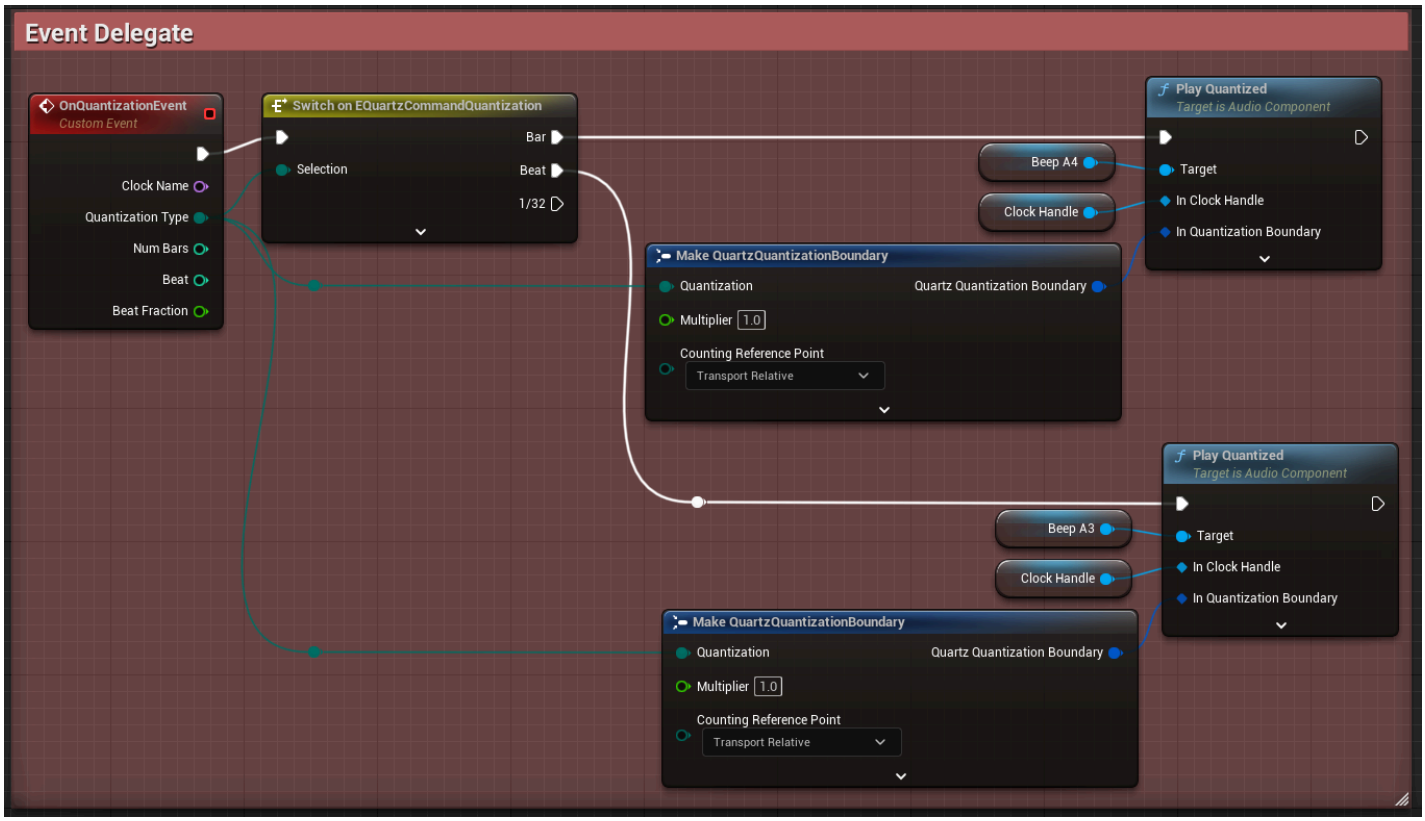
1. On the **Set Beats Per Minute** node (at the end of the previous section), drag off the **Exec Output (>)** pin and create a **Create Sound 2D** node.
2. On the **Create Sound 2D** node:
    a. Set **Sound** to `MSS_BeepA4`.
    b. Drag off the **Return Value** pin and select **Promote to Variable**. This creates a **Set** node and another Blueprint variable named `NewVar`.
    c. Drag off the **Audio Component Output** pin and create a **Set Play Multiple Instances** node.
    d. Connect the **Exec Output (>)** pin to the **Set Play Multiple Instances** node.
3. In the **My Blueprint** panel, right-click the **NewVar** variable and select **Rename**.
4. Name the variable `BeepA4`.
5. On the **Set Play Multiple Instances** node:
    a. Enable **Play Multiple Instances**.
    b. Drag off the **Exec Output (>)** pin and create a second **Create Sound 2D** node.
6. On the second **Create Sound 2D** node:
    a. Set **Sound** to `MSS_BeepA3`.
    b. Drag off the **Return Value** pin and select **Promote to Variable**. This creates a **Set** node and another Blueprint variable named `NewVar`.
    c. Drag off the **Audio Component Output** pin and create a second **Set Play Multiple Instances** node.
    d. Connect the **Exec Output (>)** pin to the **Set Play Multiple Instances** node.
7. In the **My Blueprint** panel, right-click the **NewVar** variable and select **Rename**.
8. Name the variable `BeepA3`.
9. On the second **Set Play Multiple Instances** node:
    a. Enable **Play Multiple Instances**.

2.3 - Start the Quartz Clock



1. Right-click in an empty space and add a **Get Clock Handle** node.
2. On the **Get Clock Handle** node, drag off the output pin and create a **Start Clock** node.
3. On the **Start Clock** node:
    a. Connect the **Exec Input (>)** pin to the second **Set Play Multiple Instances** node (at the end of the previous section).
    b. Drag off the **Clock Handle** node and create a **Subscribe to All Quantization Events** node.
    c. Connect the **Exec Output (>)** pin to the **Subscribe to All Quantization Events** node.
4. On the **Subscribe to All Quantization Events** node, drag off the **On Quantization Event** pin and create a **Create Event** node.
5. On the **Create Event** node:
    a. Click the dropdown and select **[Create a matching event]**. This creates a new **Custom Event** node.
    b. Name the **Custom Event** `OnQuantizationEvent`.

2.4 - Construct the Event Delegate



1. On the **OnQuantizationEvent** node:
    a. Drag off the **QuantizationType** pin and create a **Switch on EQuartzCommandQuantization** node.
    b. Connect the **Exec Output (>)** pin to the **Switch on EQuartzCommandQuantization** node.
    c. Drag off the **QuantizationType** pin and create a **Make QuartzQuantizationBoundary** node.
    d. Drag off the QuantizationType pin and create a second **Make QuartzQuantizationBoundary** node.
2. On both of the **Make QuartzQuantizationBoundary** nodes, set the **Counting Reference Point** to Transport Relative.
3. On the **Switch on EQuartzCommandQuantization** node:
    a. Drag off the **Bar** pin and create a **Play Quantized** node.
    b. Drag off the **Beat** pin and create a second **Play Quantized** node.
4. On the first **Play Quantized** node:
    a. Drag off the **Target** pin and create a **Get Beep A4** node.
    b. Drag off the **In Clock Handle** pin and create a **Get Clock Handle** node.
    c. Connect the **In Quantization Boundary** pin to the output pin of one of the **Make QuartzQuantizationBoundary** nodes.

5. On the second **Play Quantized** node:
    a. Drag off the **Target** pin and create a **Get Beep A3** node.
    b. Drag off the **In Clock Handle** pin and create a **Get Clock Handle** node.
    c. Connect the **In Quantization Boundary** pin to the output pin of the unconnected **Make QuartzQuantizationBoundary** node.
6. Compile and save the Blueprint.
7. Close the **Blueprint Editor**.

2.5 - Test the Level

Click the **Play** button on the **Level Editor Toolbar**. The low-frequency beep MetaSound plays every beat, and the high-frequency beep MetaSound plays every bar.

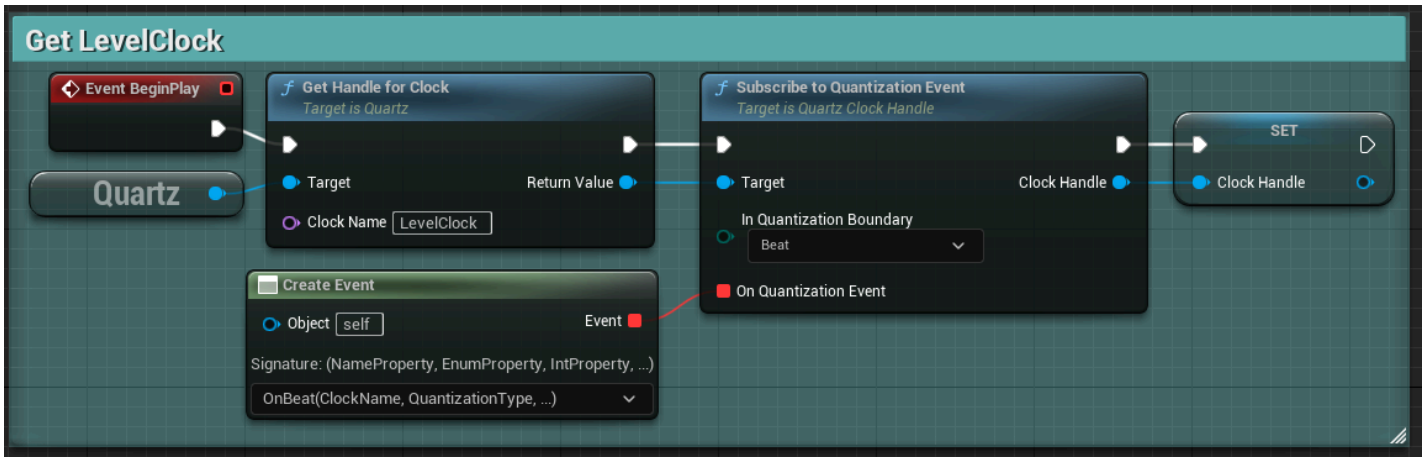
3 - Build the Blueprint Actor

Create a Blueprint Actor with a Cube Component that scales with the beat of the Quartz Clock that you set up on the Level Blueprint.

3.1 - Create the Blueprint Actor

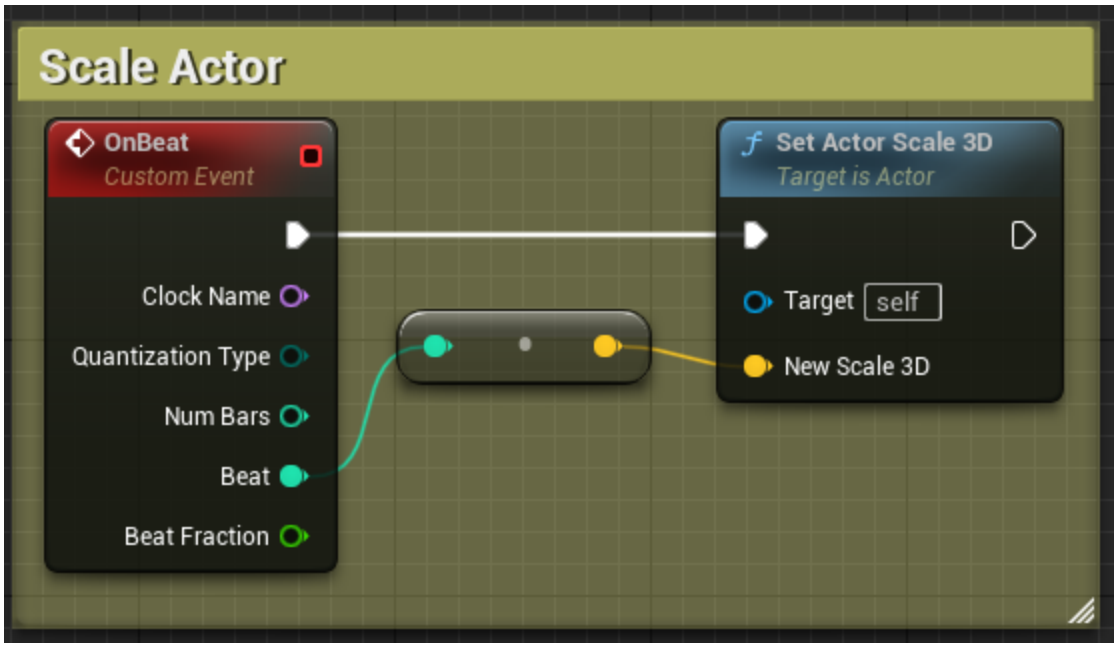
1. In the **Content Browser**, click the **Add** button.
2. Select **Blueprint Class**.
3. From the **Pick Parent Class** window, select **Actor**.
4. Name the new Blueprint Actor `BP_QuartzCube`.
5. Double-click the Blueprint Actor to open the **Blueprint Editor**.
6. In the **Components** panel, click the **Add** button, type "Cube" into the search bar, and press Enter.

3.2 - Get the Level Clock on the Actor's Event Graph



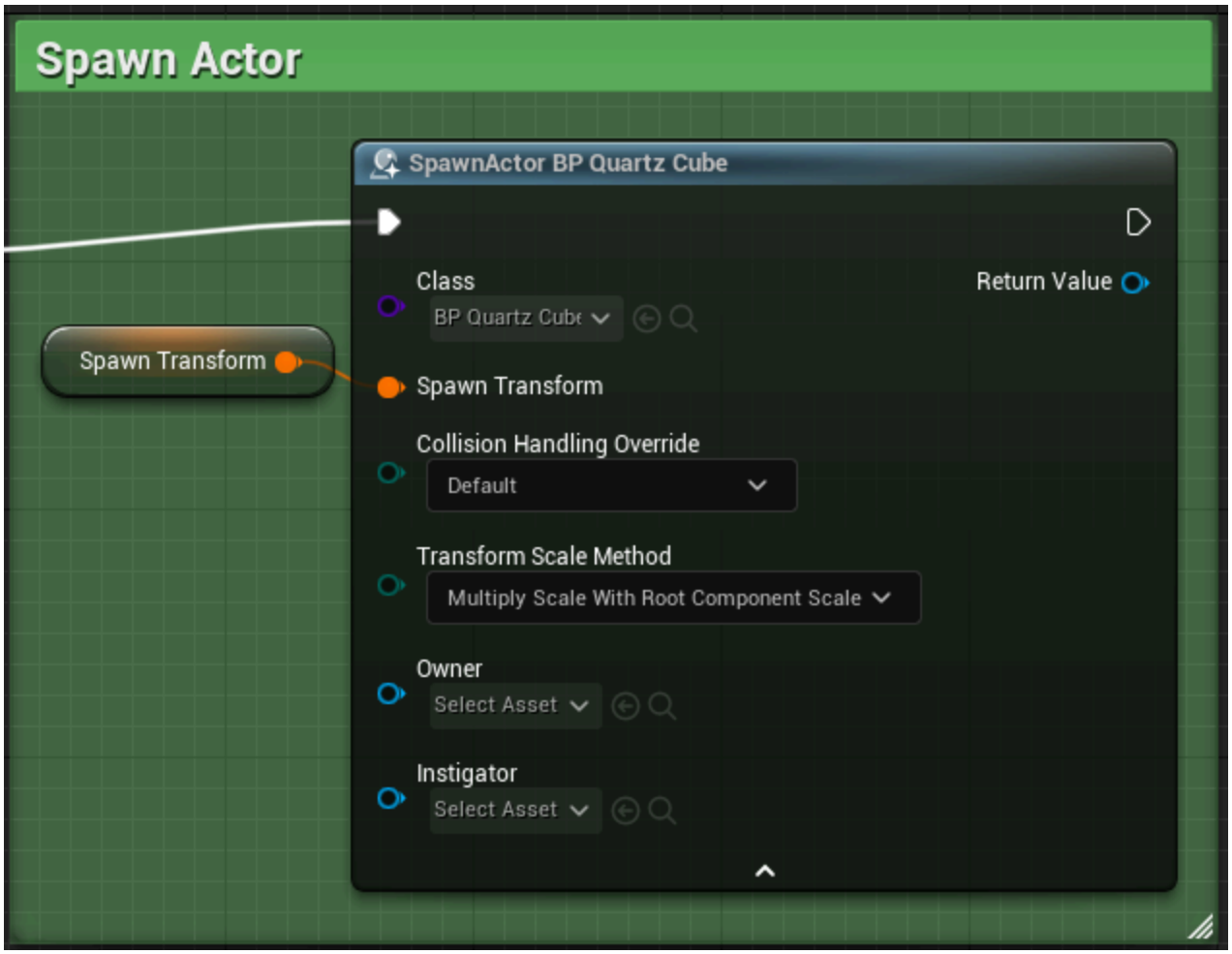
1. Right-click in an empty space and add a **Get QuartzSubsystem** node.
2. On the **Get QuartzSubsystem** node, drag off the output pin and add a **Get Handle for Clock** node.
3. On the **Get Handle for Clock** node:
    a. Connect the **Exec Input (>)** pin to the **Event BeginPlay** node.
    b. Set **Clock Name** to `LevelClock`.
    c. Drag off the **Return Value** pin and create a **Subscribe to Quantization Event** node.
    d. Connect the **Exec Output (>)** pin to the **Subscribe to Quantization Event** node.
4. On the **Subscribe to Quantization Event** node:
    a. Set **In Quantization Event** to Beat.
    b. Drag off the **Return Value** pin and select **Promote to Variable**. This creates a Blueprint variable named `Clock Handle` to store the Quartz Clock Handle and prevent its' garbage collection.
    c. Connect the **Exec Output (>)** pin to the **Set (Clock Handle)** node.
    d. Drag off the **On Quantization Event** pin and create a **Create Event** node.
5. On the **Create Event** node:
    a. Click the dropdown and select **[Create a matching event]**. This will create a new **Custom Event** node.
    b. Name the **Custom Event** `OnBeat`.

3.3 - Scale the Actor on Beat



1. On the **OnBeat** node:
    a. Drag off the **Exec Output (>)** pin and create a **Set Actor Scale 3D** node.
    b. Connect the **Beat** pin to the **New Scale 3D** pin on the **Set Actor Scale 3D** node. This automatically creates a **To Vector (Integer)** node.
2. Compile and save the Blueprint.
3. Close the **Blueprint Editor**.

4 - Modify the Level Blueprint



Add logic to the Level Blueprint to spawn the `BP_QuartzCube`.

1. On the **Level Editor Toolbar**, click the **Blueprint** button and select **Open Level Blueprint**.
2. On the **Subscribe to All Quantization Events** node (in section 2.3), drag off the **Exec Output (>)** pin and create a **Spawn Actor from Class** node.
3. On the **Spawn Actor** node:
    a. Set **Class** to `BP_QuartzCube`.
    b. Drag off the **Spawn Transform** pin and select **Promote to Variable**.
4. Compile the Blueprint.
5. Select the **Spawn Transform** node.
6. In the **Details** panel, set the **Default Value > Spawn Transform > Location** to 1600.0, 1200.0, 200.0.
7. Compile and save the Blueprint.
8. Close the **Blueprint Editor**.

5 - Test the Level

Click the **Play** button on the **Level Editor Toolbar**. The `BP_QuartzCube` now spawns in the level and scales with the metronome beat.