

## Audio Reactive SHADERS



# Welcome to Audio Reactive Shaders Documentation

Thank you for your interest in **Audio Reactive Shaders**!

This documentation is designed to help you get the most out of your purchase, whether you're a seasoned developer or just starting out with Unity.

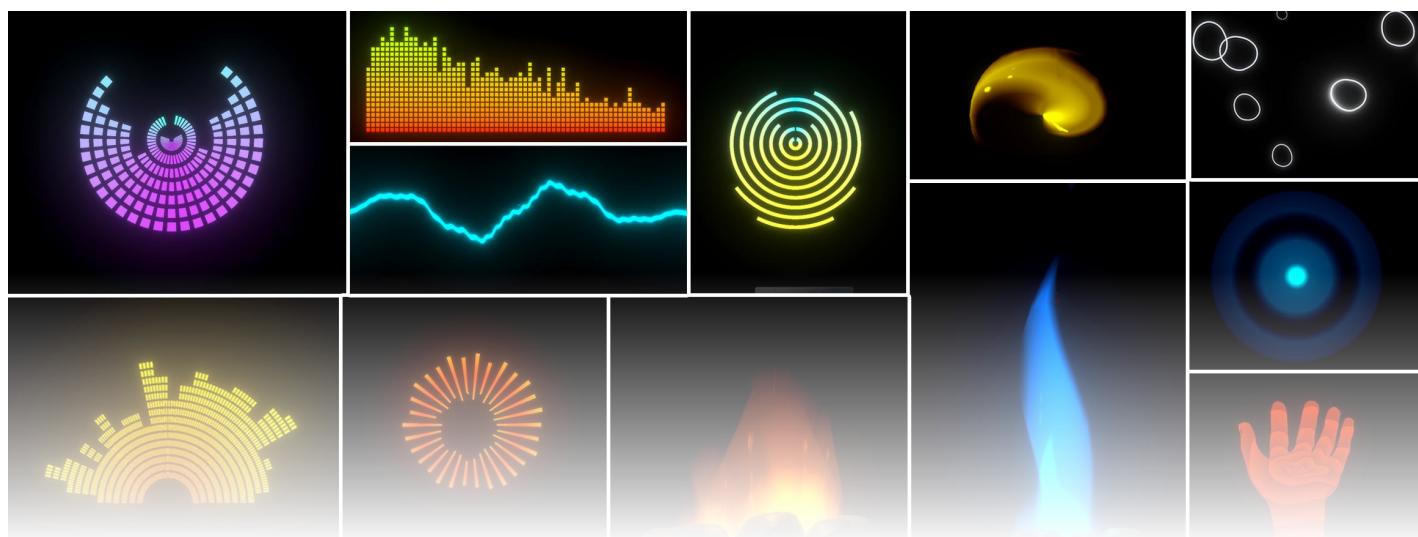
This asset allows you to **integrate materials in your project that react in real-time to audio**, bringing an extra layer of immersion and excitement to your projects.

In this guide, you'll find detailed **instructions, examples, and tips** to seamlessly integrate and customize Audio Reactive Shaders in your Unity environment. Whether you're looking to enhance a game, create an interactive experience, animation, or experiment with new artistic expressions, our shaders offer endless possibilities.

I **highly value your review** in the Asset Store as it plays a crucial role in the ongoing development and improvement of Audio Reactive Shaders. **Positive feedback** not only helps other users understand the benefits of our asset but also helps me improve and further develop it.

I hope this documentation answers all your questions and inspires you to push the boundaries of what's possible with audio-reactive visuals. Should you need any further assistance, **don't hesitate to reach me out**.

Happy creating!



# ⚡ Quick-start guide

On the example project you will be able to see all the shaders in action but lets make a quick start from scratch. We will use as example a project using the URP pipeline.

The Audio Reactive Shaders needs **4 main components** to work:

- 1. An audio source**
- 2. The Music Spectrum Reader component**
- 3. An audio reactive material of your choice**
- 4. An Audio Data Interpreter component (\*)**



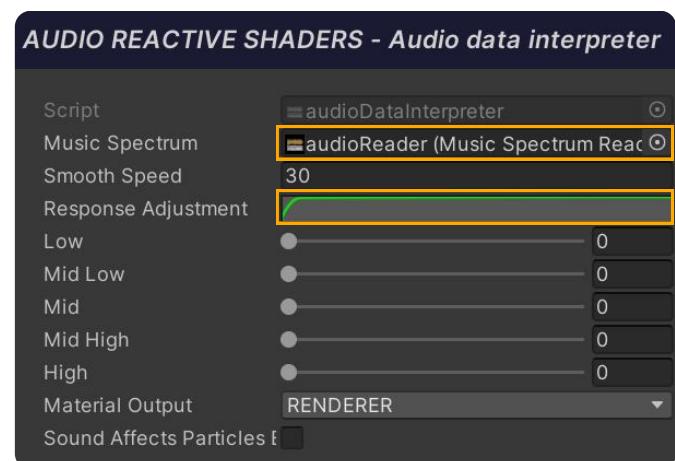
**Tip:** I would recommend to group the **audio source + audio reader** in one game object and **audio interpreter + material** applied to another object to improve scalability and maintenance.

Now create a game object, add an **audio source** with your **audio file** inside and add the **Music Spectrum Reader component**. <sup>(1)</sup>

Then add a 2d or 3d shape, add an audio reactive material <sup>(2)</sup> to it, for example fire, add an **Audio Interpreter component** <sup>(1)</sup> and drag the object with the **Music Spectrum Reader** on the **Music spectrum field** of the Audio Interpreter component.

Then on the **audio response curve** choose an ascending preset (like the one you see on the image).

For this quick start we will leave all the other variables as default.



## Ready to go!

Play the scene and see the shader in action.

In the next sections we will see the details, the options you have with **Audio Reactive Shaders** and how to use it in **2d, 3d objects, particles or UI elements**.

(\*) depending on the material you choose you might need a regular or a dynamic audio interpreter, see your chosen shader name to know which one to use. More details in the **Audio Interpreter component** section

(1) included on the scripts folder

(2) included on the materials folder

## How it works

As the **audio source** plays an audio file the **Music Spectrum Reader** will read de audio data and transform it to numbers divided into frequency bands.

Then the **Audio Data Interpreter** will take this raw data and adapt it so it can be personalized and sent to the chosen **material** to finally be visualized on your game object. **All the shaders and materials are procedural**, that means that you can scale or stretch them as much as you want without any pixelation and they dont need much hard drive space.

## Main components:

In this package you will find 2 scripts that makes the system work, you dont need to write any code, just drag and drop them into the scene as described in the Quick start guide.

### **Music Spectrum Reader:**

This component reads de audio data and transforms it to numbers divided into frequency bands.

**Input selection:** Here you can chose between a specific **audio source** (should be attached to the same game object)

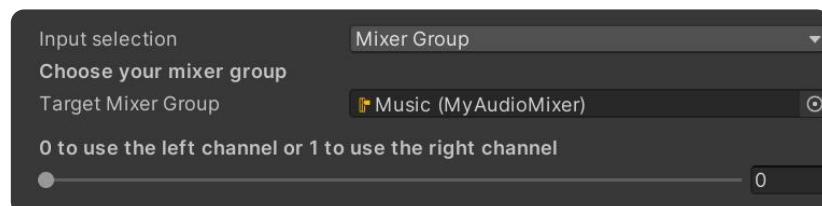
Use **Audio Listener** to make the materials react to all the sounds the player can hear.

**Mixer Group** will let you choose a group of the audio mixer and react to the audio sources that are played there.

**Web GL** versions adds compatibility to export for WebGL.

**The Channel slider** lets you select between left o right channel of a stereo audio file.

**Show hidden vars** let's you look in real time how the component is working on the inside.



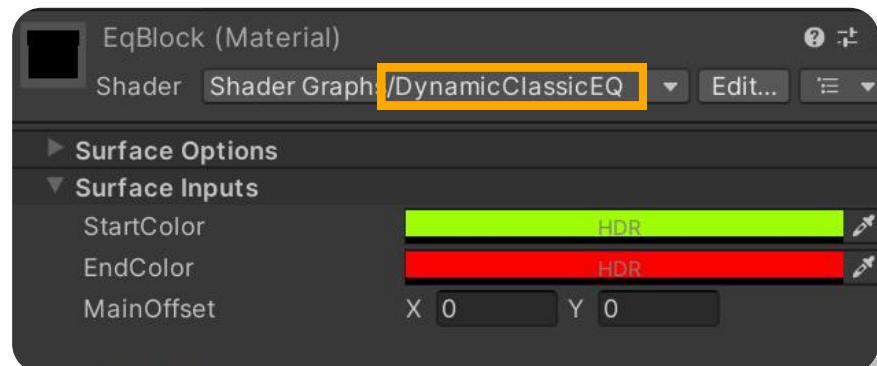
### Tips:

- You can have multiple audio readers working with different audio inputs and send each one to different materials.
- If you need to refresh de audio mixer group list you can call the `refreshAudioSourcesOnMixerGroup()` method on the **Music Spectrum Reader** script or disable and re-enable the object containing this script.

## Audio Data Interpreter:

This component takes the raw data and adapts it so it can be personalized and sent to the chosen material. There are 2 types of Audio data interpreters, regular and dynamic.

The regular is for materials that needs 5 frequency bands and the dynamic for materials that can use from 1 up to 64 bands to visualize.



You will know which one to choose because the material shaders that require dynamic audio data interpreters have 'Dynamic' in their names. The others use the regular shader.

### Audio Data Interpreter (regular):

**Music Spectrum** Drop the object containing the Spectrum Reader you want this shader to react to here.

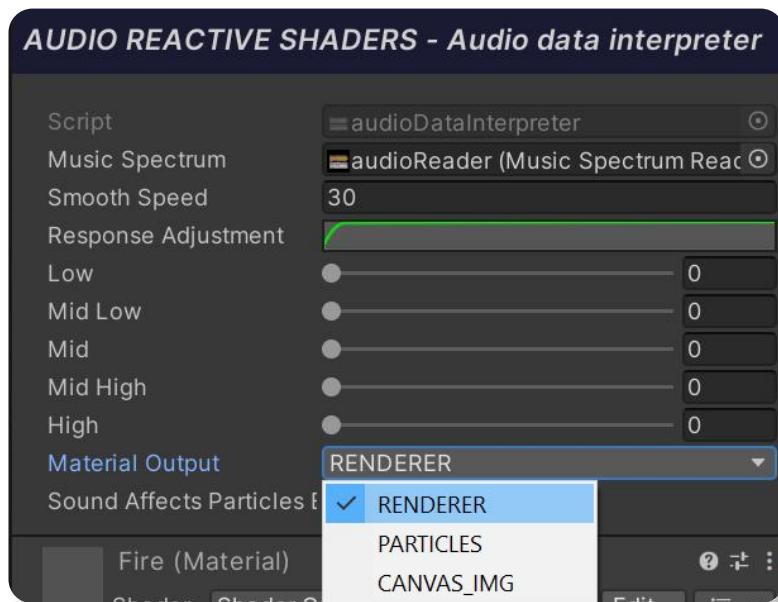
**Smooth Speed** higher values will make the animation react faster while lower values will make it smoother. Leave it in 0 to remove the smooth completely.

**Response Adjustment** determines how the material is going to react along the sound intensity range, a logarithmic curve works well in most cases. (see the image)

**The Frequency Sliders** lets you visualize how this shader will process the data.

**Material Output** allows you to chose where the material will be displayed. Renderer for regular 3d objects, Particles or canvas\_img for UI elements

**Sound Affects Particles emission** makes the sound affect the emission rate, (more sound more particles.)



### Dynamic Audio Data Interpreter:

**Music Spectrum** Same as the regular data interpreter.

**Smooth Speed** Same as the regular data interpreter.

**Response Adjustment** Same as the regular data interpreter.



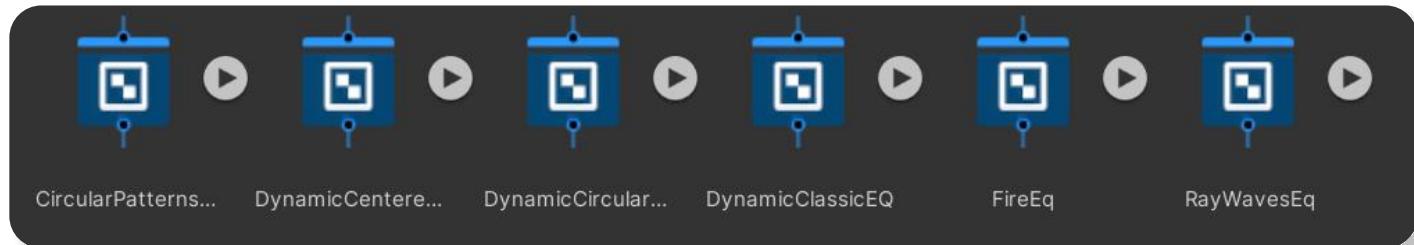
**Bands** lets you chose how many bands you want to visualize from 1 to 63. That is the main difference between Dynamic and regular shaders.

**Material Output** Same as the regular data interpreter.

**Sound Affects Particles emission** Same as the regular data interpreter.

# The Shaders!

In this package you will find 6 audio reactive shaders. 3 regulars and 3 dynamics with which you can create many different materials or use the included ones.



## DynamicClassicEQ

Lets you create classic or round Eq visualizers with up to 63 bands to visualize and many options to personalize it.



**EndColor and BaseColor** Lets you create a gradient from the base to the top of the Eq.

**Subdivisions/BorderWidth:** Lets you select the separation between bands and subdivisions.

**Subdivisions/Subdivisions:** Lets you subdivide the bands on the X and Y axis.

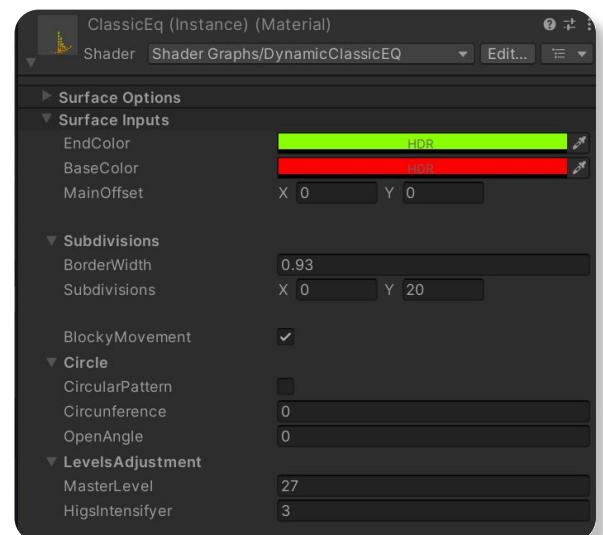
**Subdivisions/BlockyMovement:** Makes the animation fit in the subdivided cells.

**Circle/CircularPattern:** Makes the Eq follow a circular shape.

**Circle/Circumference:** Controls the circle size.

**Circle/OpenAngle:** Opens an angle below the circle.

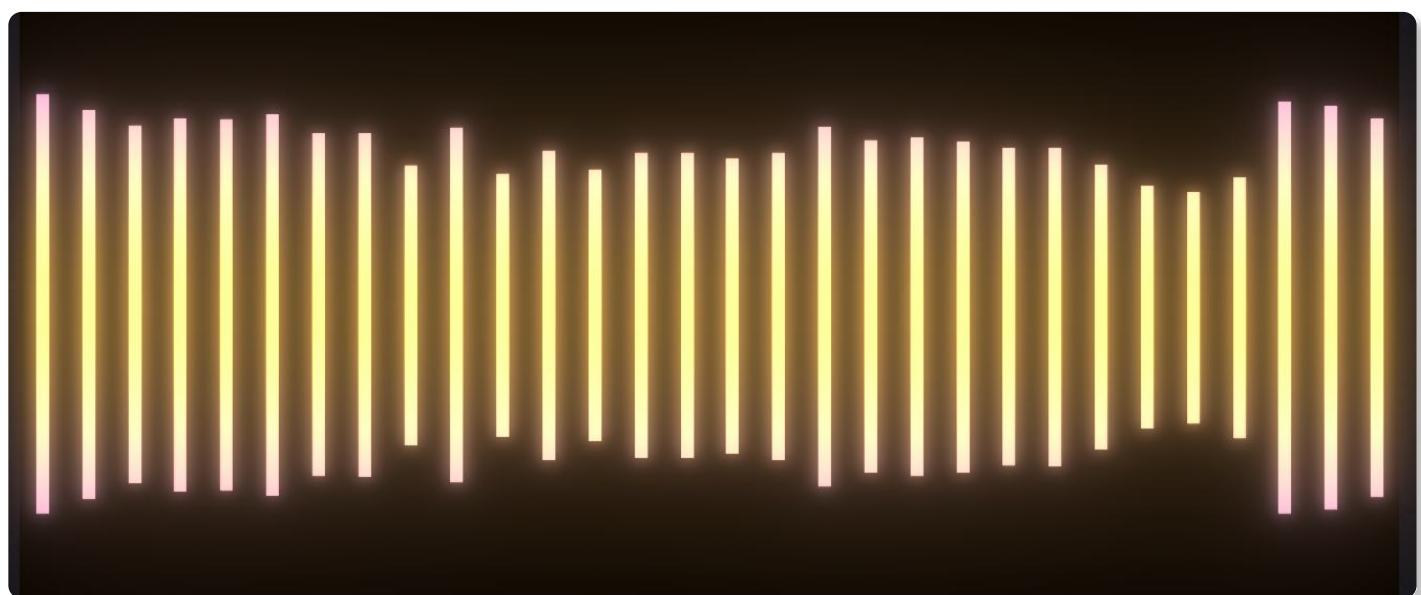
**LevelsAdjustment:** lets you control the master level and boost the high frequencies if needed to show a more even visualization.



# The Shaders!

## DynamicCenteredEQ

Lets you create modern Eq visualizers that expands on the Y axis with the sound. You can set up to 63 bands to visualize and many options to personalize it.



**EndColor and BaseColor** Lets you create a gradient from the base to the top of the Eq.

**Offset:** Controls the main offset of the EQ.

**Subdivisions/BorderWidth:** Lets you select the separation between bands and subdivisions.

**Subdivisions/Subdivisions:** Lets you subdivide the bands on the X and Y axis.

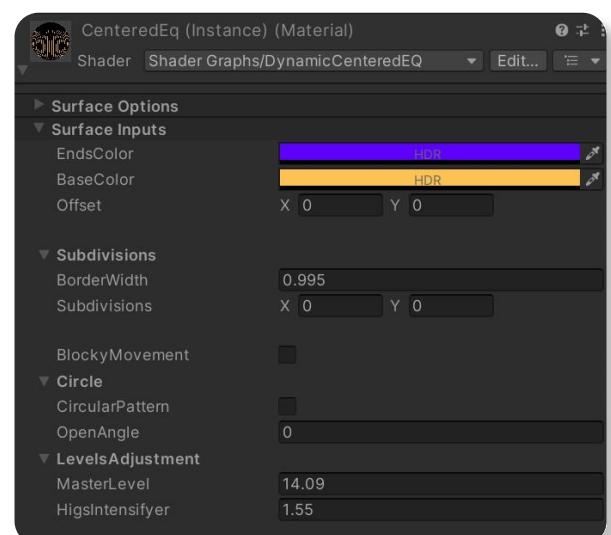
**Subdivisions/BlockyMovement:** Makes the animation fit in the subdivided cells.

**Circle/CircularPattern:** Makes the Eq follow a circular shape.

**Circle/Circumference:** Controls the circle size.

**Circle/OpenAngle:** Opens an angle below the circle.

**LevelsAdjustment:** lets you control the master level and boost the high frequencies if needed to show a more even visualization.



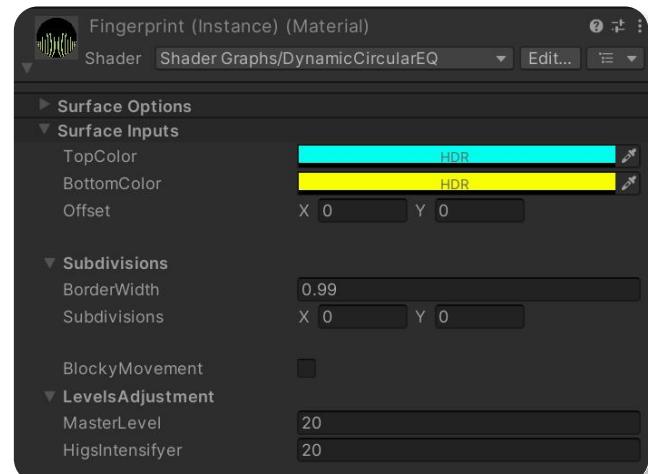
# The Shaders!

## DynamicCircularEQ

This shader divides the frequency bands in circles and expands on the circumference of each band, with up to 63 bands to visualize and many options to personalize it.



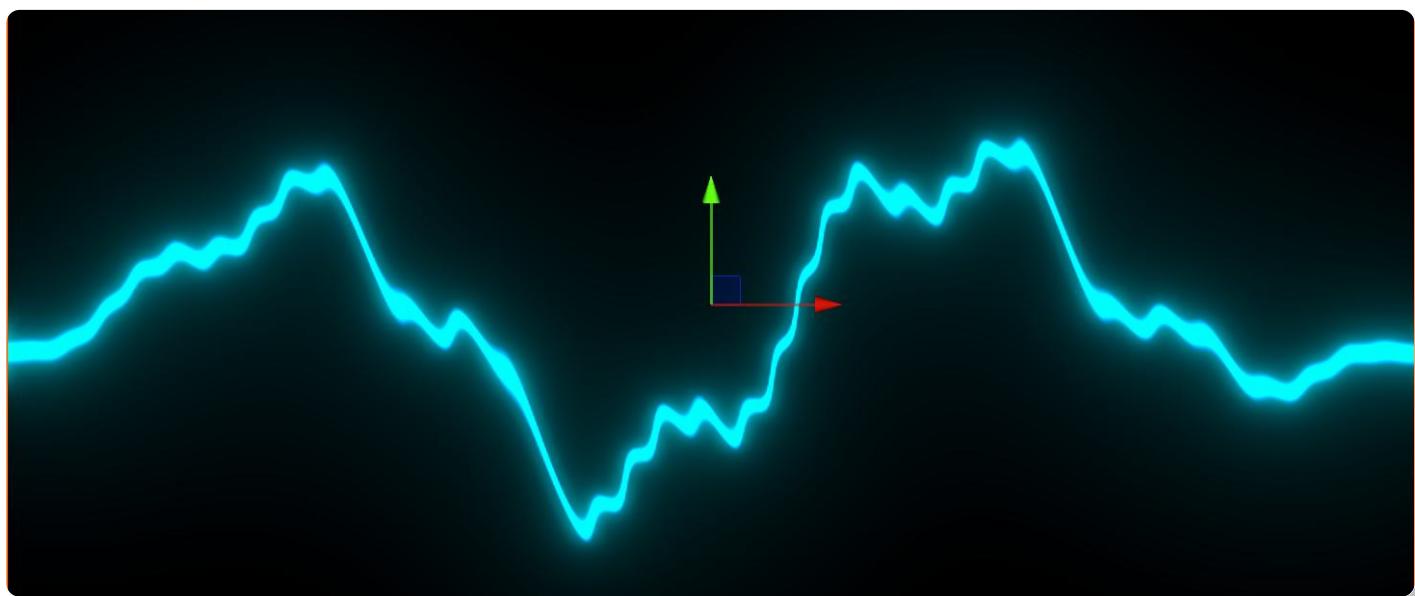
**EndColor and BaseColor** Lets you create a gradient from the base to the top of the Eq.  
**Offset:** Controls the main offset of the EQ.  
**Subdivisions/BorderWidth:** Lets you select the separation between bands and subdivisions.  
**Subdivisions/Subdivisions:** Lets you subdivide the bands on the X and Y axis.  
**Subdivisions/BlockyMovement:** Makes the animation fit in the subdivided cells.  
**LevelsAdjustment:** lets you control the master level and boost the high frequencies if needed to show a more even visualization.



# The Shaders!

## RayWavesEQ

Is a waveform visualizer tuned up so it can looks like electricity rays. The start and end points are defined so you could use it in guns with a line tenderer for example.



**CenterColor** and **BorderColor** Lets you create a gradient from the base to the top of the Eq.

**RayWidth** controls the width of the ray

**Heiht-Circumference** Controls the Y offset or the diameter of the circumference, depending of the activation of the circular pattern.

**WaveDisplacement** controls if the wave is going to be offsetting over time on the X axis.

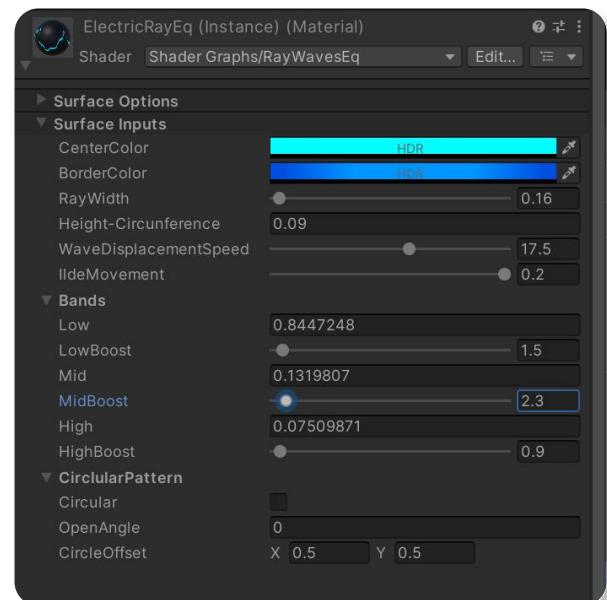
**IldeMovement** creates a base wave displacement that runs without sound.

**Bands** lets you take a look at how the 3 frequencys are being received and accentuate the ones you want. Low affects the bigger wave, mid the middle ones and high the small waves inside.

**Circle/CircularPattern** Makes the Eq follow a circular shape.

**Circle/OpenAngle** Opens an angle bellow the circle.

**CircleOffset** lets you offset the circle on the X and y axis.



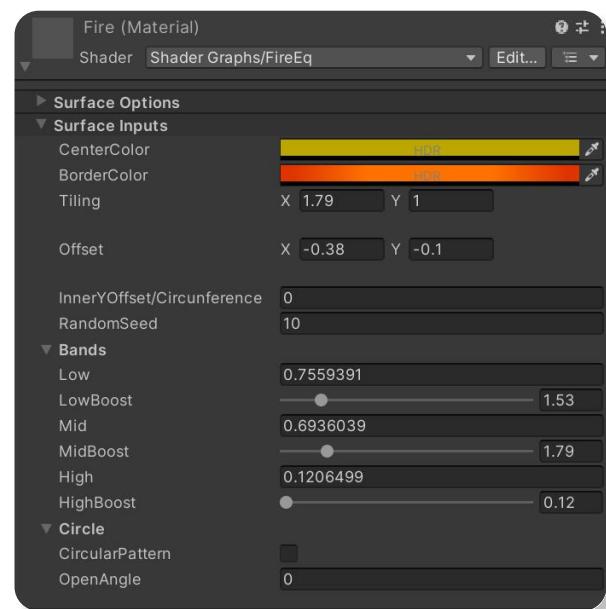
# The Shaders!

## FireEQ

Is a procedurally generated stylized fire that reacts and grows with the sound



**CenterColor and BorderColor** Lets you create a gradient from the base to the top of the Eq.  
**Tiling and Offset** lets you place and scale the fire in the X and Y axis  
**InnerYOffset-Circumference** Controls the internal Y offset or the diameter of the circumference, depending of the activation of the circular pattern.  
**RandomSeed** lets you change the shape of the fire, especially useful if you are combining 2 or more and you want them to look different.  
**Bands** lets you take a look at how the 3 frequencys are being received and accentuate the ones you want. Low affects the bigger wave, mid the middle ones and high the small waves inside.  
**Circle/CircularPattern** Makes the Eq follow a circular shape.  
**Circle/OpenAngle** Opens an angle bellow the circle.



# The Shaders!

## CircularPatternsEQ

Is a circles pattern that deforms with the music taking flower like shapes.



**Color** Lets you chose the main color.

**CircleSize** Lets you change the circumference of the circles.

**Tiling and Offset** lets you place and scale the fire in the X and Y axis.

**Bands** lets you take a look at how the 4 frequencies this shader takes, are being received and accentuate the ones you want. Higher normalizing numbers will result in a more accentuated movement of the texture.





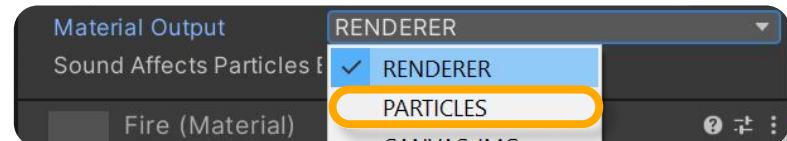
# Material options

## 2d or 3d game objects

To use Audio reactive shaders just follow the quick start guide and adapt the material to you needs

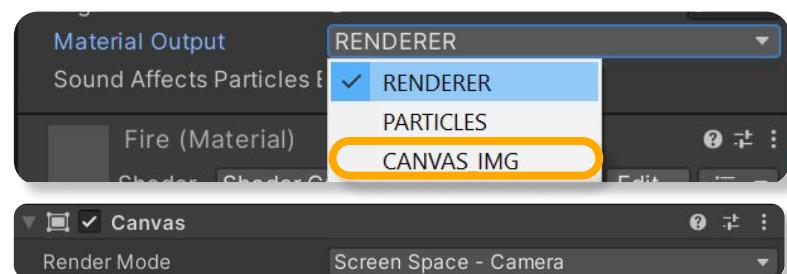
## Particles

To use with particles choose and audio reactive material for the material field of the particles renderer, add the Audio data interpreter and then choose PARTICLES in the Material Output.



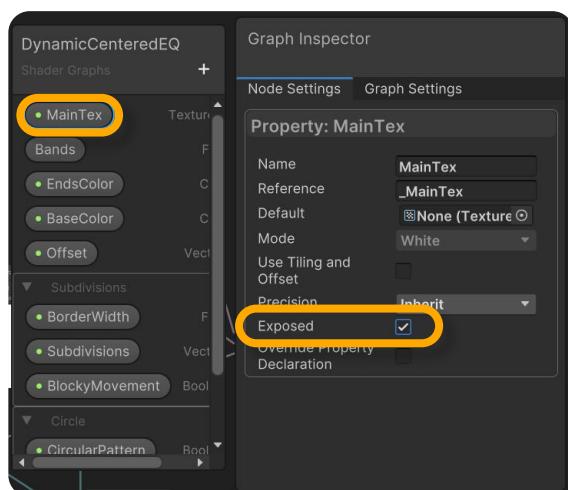
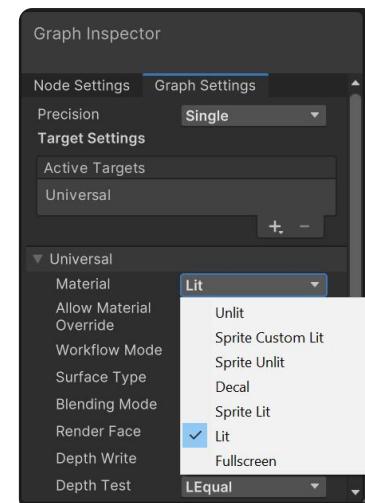
## UI-Canvas

To use it in a canvas UI use the audio reactive material in a image component of your canvas, add the Audio data interpreter and then choose CANVAS\_IMG in the Material Output. Then **set up the canvas as screen space camera** as render mode.



**If you need the canvas in overlay** then you can go to the shader graph window of the shader you are using on the graphs settings change the shader **from unlit to lit**.

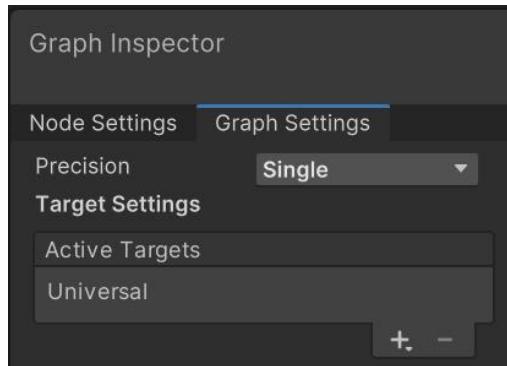
The only downside is that the material wont be affected with post processing volumes. But the shader itself will work normally with the colors you choose.



**Tip for UI-Canvas and 2D sprites:** You might see a warning log saying that the image doesn't find a property called "MainTex". to get rid of it open to the shader you are using, expose the "MainTex" property, save it and play the scene now the warning will be gone and you can unexpose the "MainTex" again.

## About the render pipelines compatibility.

This system works with the Built in, HD and URP render pipelines. By default it is using URP due to its flexibility and performance in comparison with the others. If you want to use it with another render pipeline you just have to convert the shaders by changing the active target to the one you want to use and removing the URP (Otherwise the materials will look pink).



If also you want to use the demo scene with other render pipelines you may want to remove the URP scripts on the lights and post-Volume to get rid of the warnings.

### V 1.1.0

This update adds **WebGL compatibility**.

### V 1.2.0 (current)

This update adds **compatibility with the audio mixer** as audio input.



\*\*\*\*\*

If you like the product please remember to leave a positive review on the asset store.

Discord:

<https://discord.gg/NPTTQd6ZZm>

Here you can take a look at my other assets :)

<https://assetstore.unity.com/publishers/77287>

#### Terms of use

- You may use this asset in your personal or commercial projects.
- You may not sell this asset or modify it in any way to resell it as similar asset.
- The customer support will be available as long as the source code is not modified. Anyways the code is documented so intermediate and advanced users will be able to modify it with ease if needed.