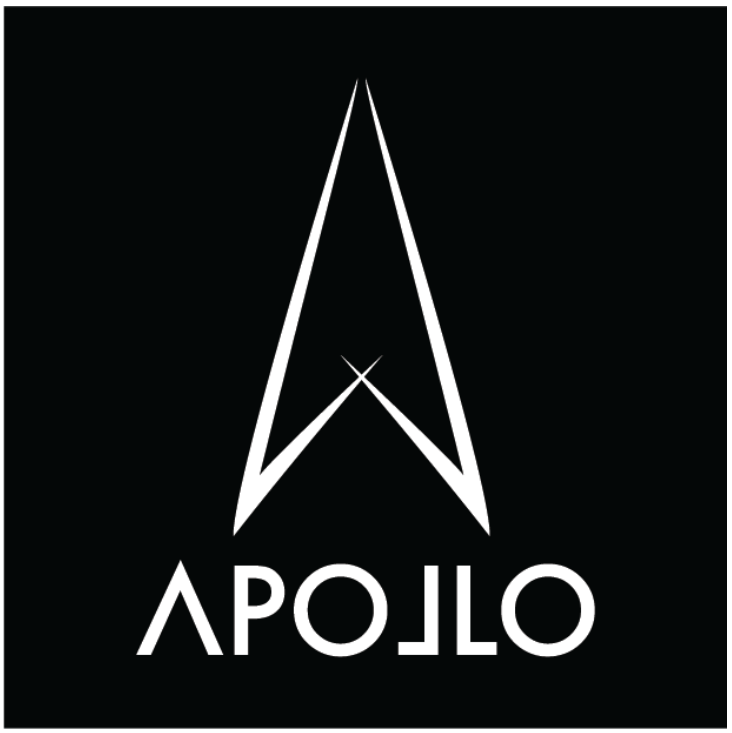


Apollo Visualizer Kit

Beta 0.4

Introduction

Apollo Visualizer Kit is a plugin tool for Unity 3D that lets the user read the intensity of 6 frequency ranges (Sub-bass, bass, low midrange, midrange, upper midrange and presence) or your custom frequency ranges on playing AudioSources in the scene in real time, this can be used to affect gameobjects in many ways like changing the the size, color, rotation or even emitting particles.

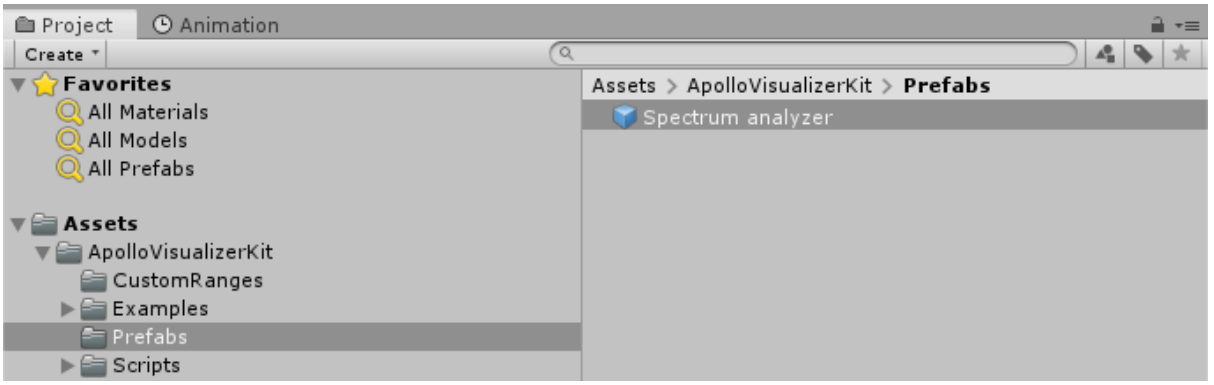
It comes with 7 built-in modifier scripts that will let you get started on making your visualizations and can serve as a starting point to build your own modifier.

You can see the online documentation here.

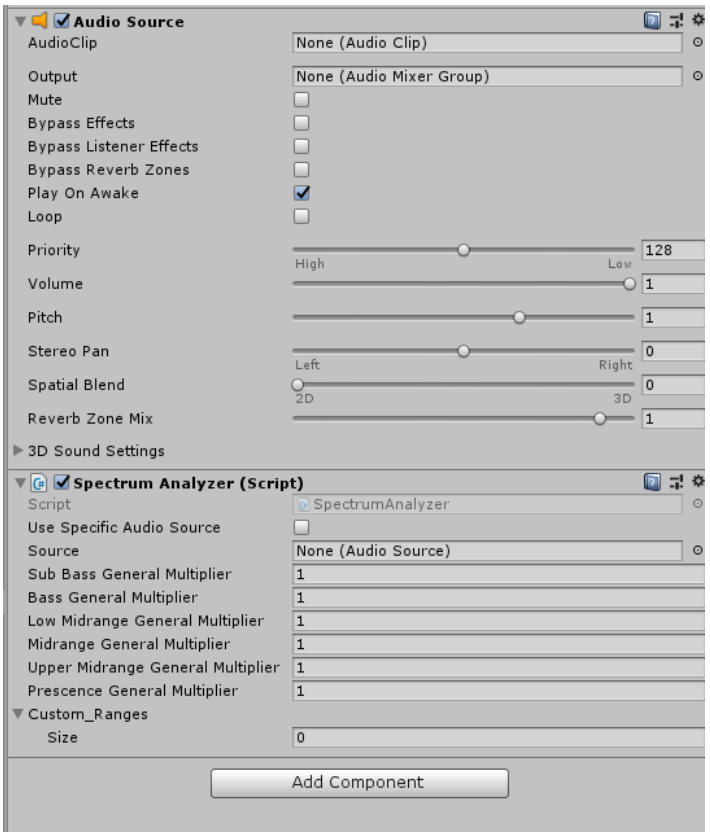
Getting Started

This is a walkthrough for you to start playing with a simple visualization.

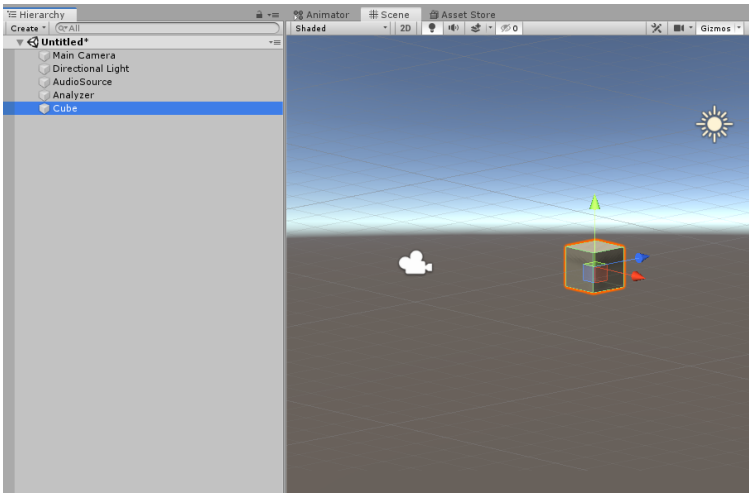
1. Open unity and start with a blank scene.(Ctrl+N)
2. Go to the prefabs folder at **Assets>ApolloVisualizerKit>Prefabs** and drag the SpectrumAnalyzer prefab to your scene.



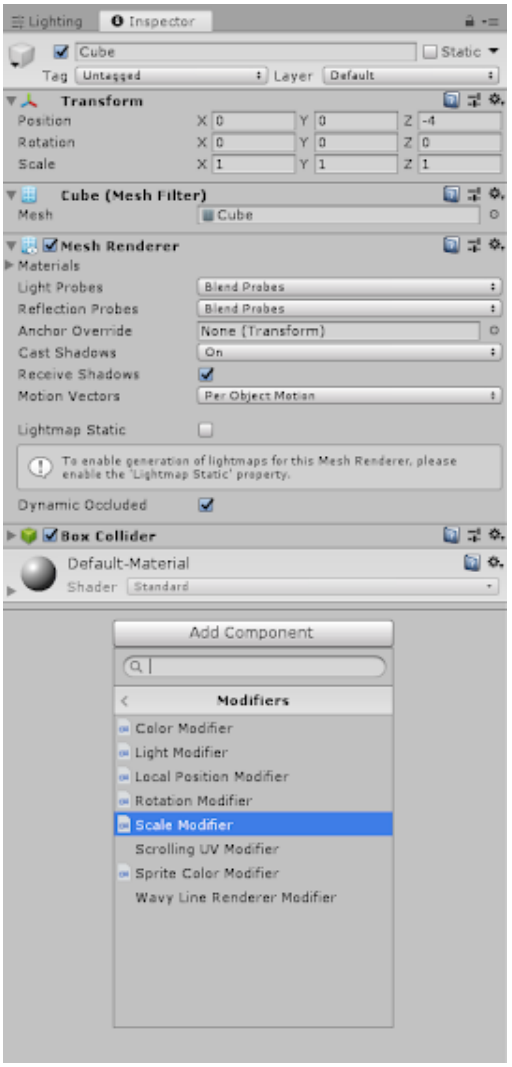
3. This prefab has been already setup with both an AudioSource and a SpectrumAnalyzer components. You can add any Audio Clips to the AudioSource component to be played on awake, this GameObject will be the one handling all the audio analysis.



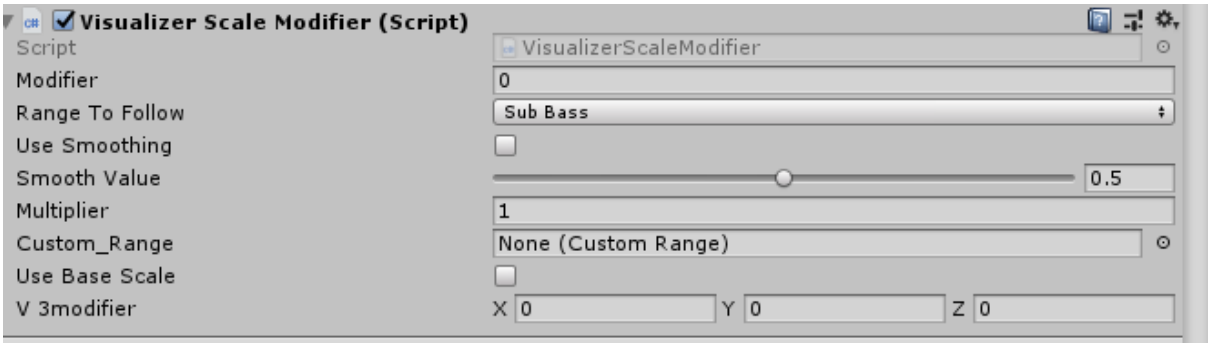
4. Now with the SpectrumAnalyzer setup we can create our objects and add modifiers to them. Let's create a cube (Right click on the hierarchy>Create>3D Object>Cube) and position it in front of your Main Camera.



5. Now let's add a Scale Modifier component to the cube (Add Component>Apollo>Modifiers>)



6. Now let's edit the values on the ScaleModifier component in the cube. Here you will see multiple values to edit:



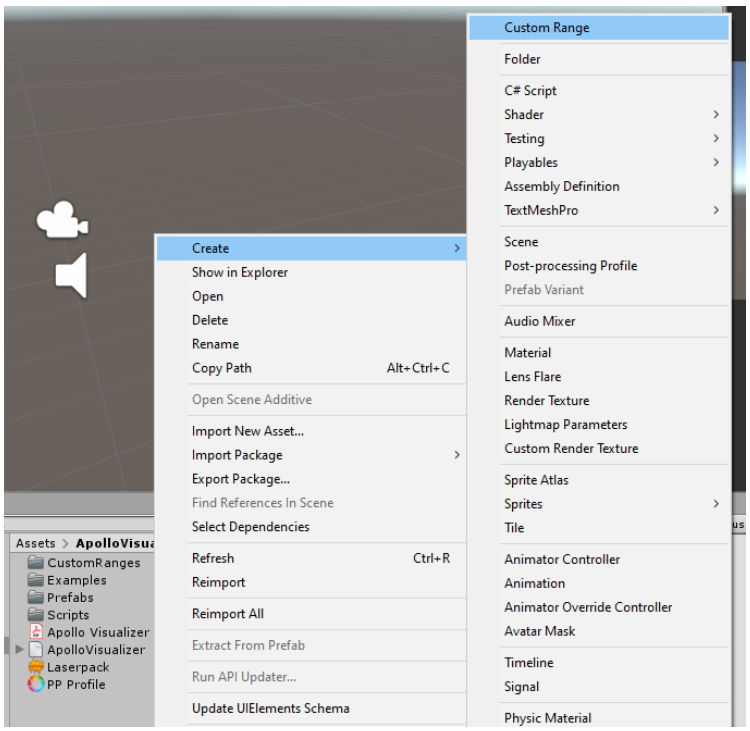
   a. **Modifier:** This value will let you know in real time what's the intensity of the frequency range. This will update on realtime, no need to change it.
   b. **Range to follow:** This dropdown will show all the frequency ranges available to follow. For now let's leave it on Midrange.
   c. **Use Smoothing:** This checkbox tells the modifier to apply lerp to the modifier value, applying a smooth effect to it. This comes included in all the scripts that inherit VisualizerObjectBase
   d. **Smooth Value:** This value determines the intensity of the smoothing, give it higher numbers to make it change faster/less smoothly. This comes included in all the scripts that inherit VisualizerObjectBase
   e. **Use Base Scale:** This checkbox lets you choose if you want the object to change size in relation to their current size or by the modifier raw value. Let's check it.
   f. **V3 Modifier:** This Vector 3 value is key in order to make the object move, it will let you choose with which intensity you want the object to change in each axis.

7. Now you're set! click play and see the cube dance.

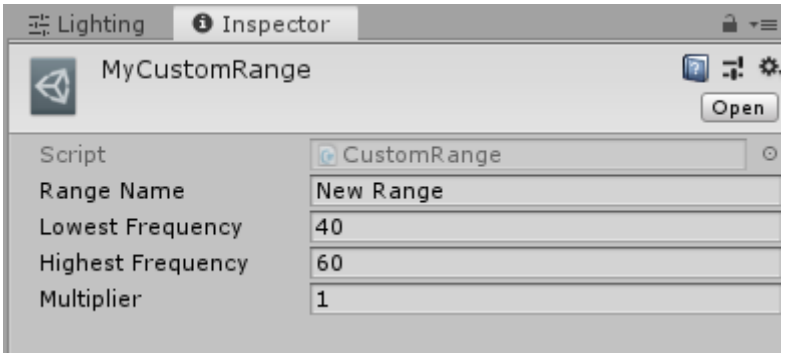
Create Custom Frequency Ranges

Sometimes the built-in frequency ranges might not be enough and you'll want to create your own presets, in order to do so we added a new Custom Range option in the create menu of the project window:

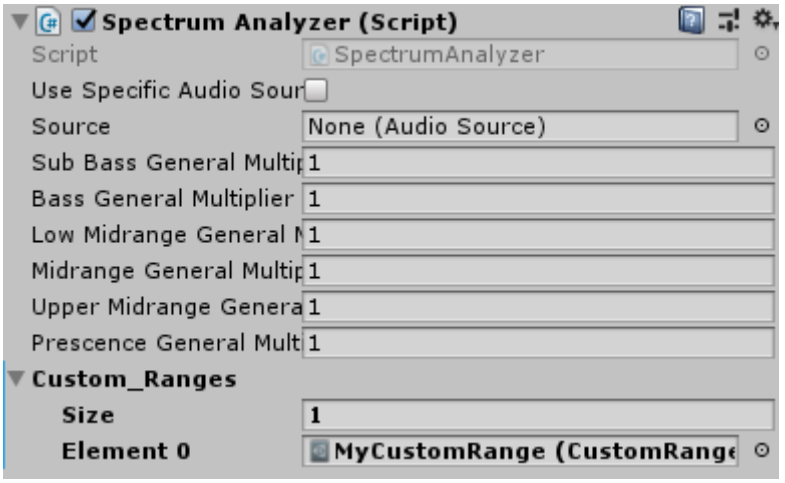
1. You can create a Custom Frequency Range by going to the create menu on your project window and select Create>Custom Range.



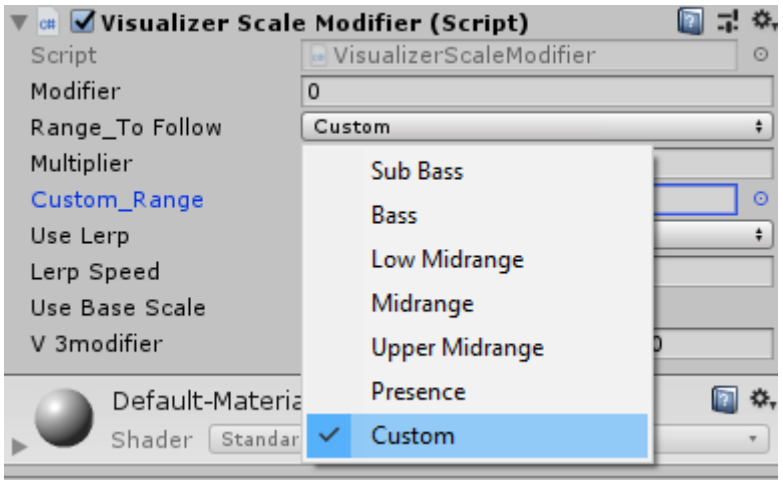
2. Name your new custom range and edit its properties.
    a. RangeName: This field sets the name of the custom range, currently is not being used.
    b. Lowest frequency: Here you add the lowest frequency in hertz you want to capture in this range.
    c. Highest Frequency: Here you add the highest frequency in hertz you want to capture in this range.
    d. Multiplier: This value is will be a global multiplier for this range modifier.



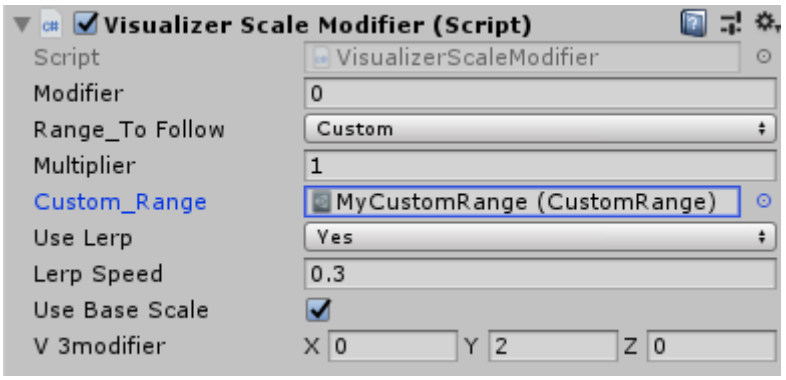
3. With the Custom Range setup ready, you now must add it to your SpectrumAnalyzer component. Select the spectrum analyzer and click the last property of the component called “Custom_Ranges”, change the size to the number of custom ranges that you’ll be using, press enter, and drag and drop your previously created Custom Ranges from the project window to their respective places.



4. Now is time to set your Modifier components. Select the object that you want to follow your custom range, click on the range to follow dropdown and select “Custom”



5. Then drag and drop your custom range object to the Custom_Range option. Now it should be ready to use your custom range.



Modifier Reference

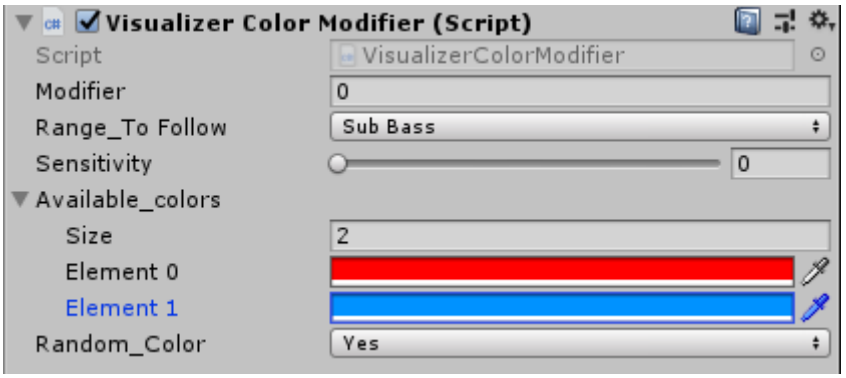
Visualizer Local Position Modifier

This scripts lets you change the local position of a GameObject according to their range to follow.

1. **Modifier:** This value will let you know in real time what's the intensity of the frequency range. This will update on realtime, no need to change it.
2. **Range to follow:** This dropdown will show all the frequency ranges available to follow.
3. **Use Base position:** This checkbox lets you choose if you want the object to change size in relation to their current size or by the modifier raw value.
4. **V3 Modifier:** This Vector 3 value is key in order to make the object move, it will let you choose with which intensity you want the object to change in each axis.

Visualizer Color Modifier

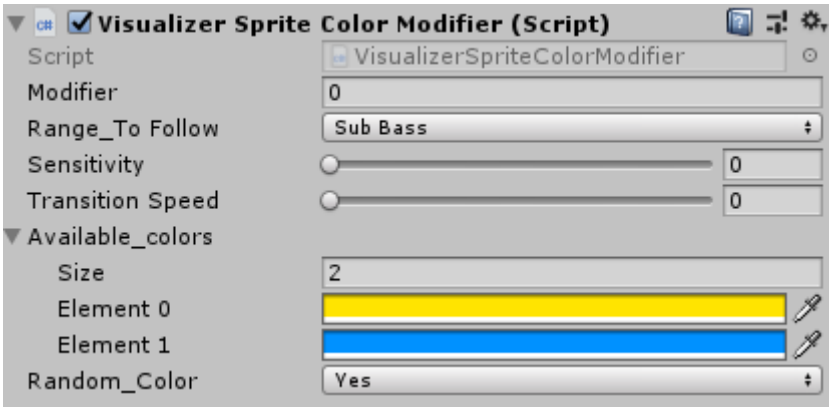
This scripts allows you to change the material.color property in your 3d object to a random color in a list or just follow it in order.



1. **Modifier:** This value will let you know in real time what's the intensity of the frequency range. This will update on realtime, no need to change it.
2. **Range to follow:** This dropdown will show all the frequency ranges available to follow.
3. **Sensitivity:** In order to mark a color change, the modifier finds differences between last modifier value and the current one. The sensitivity value lets you define how big the difference between current and last value has to be in order to trigger a color change.
4. **Available colors:** This is an array of multiple colors that you can set from the inspector.
5. **Random color:** This lets you choose whether select Available colors in order or just in a random fashion.

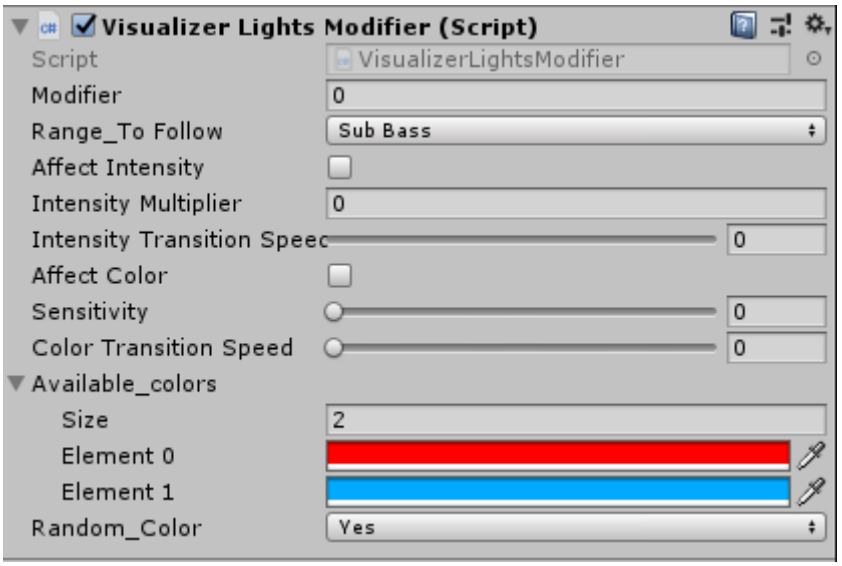
Visualizer Sprite Color Modifier

This component is made in order to control the color of a 2D sprite renderer according to the range to follow.



1. **Modifier:** This value will let you know in real time what's the intensity of the frequency range. This will update on real-time, no need to change it.
2. **Range to follow:** This dropdown will show all the frequency ranges available to follow.
3. **Sensitivity:** In order to mark a color change, the modifier finds differences between last modifier value and the current one. The sensitivity value lets you define how big the difference between current and last value has to be in order to trigger a color change.
4. **Transition speed:** This determines how fast the color changes will execute. The higher the number the snappier will be the changes.
5. **Available colors:** This is an array of multiple colors that you can set from the inspector.
6. **Random color:** This lets you choose whether select Available colors in order or just in a random fashion.

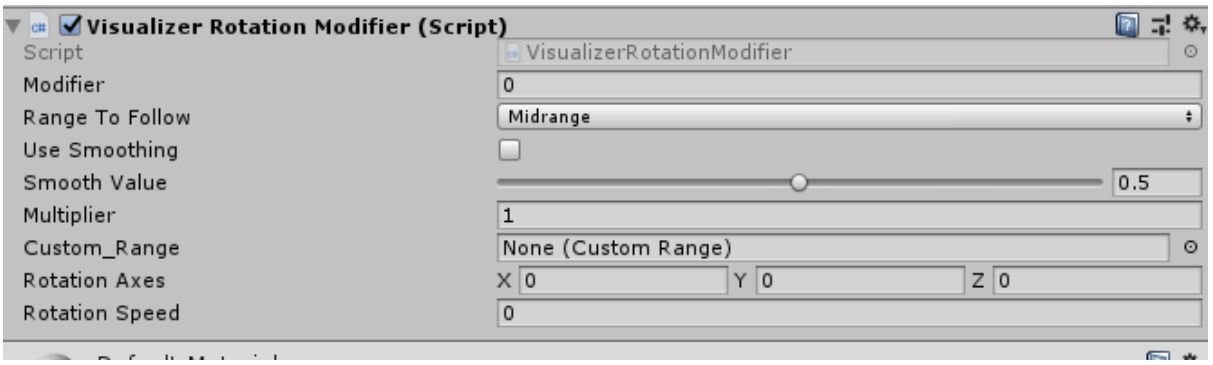
Visualizer Lights Modifier



1. **Modifier:** This value will let you know in real time what's the intensity of the frequency range. This will update on realtime, no need to change it.
2. **Range to follow:** This dropdown will show all the frequency ranges available to follow.
3. **Affect intensity:** This option turns the intensity change on the object.
4. **Intensity multiplier:** This value is multiplied by the modifier value and added to the intensity of the light.
5. **Intensity transition speed:** This determines how fast the intensity changes will execute. The higher the number the snappier will be the changes.
6. **Affect color:** This option turns on the color change on the object.
7. **Sensitivity:** In order to mark a color change, the modifier finds differences between last modifier value and the current one. The sensitivity value lets you define how big the difference between current and last value has to be in order to trigger a color change.
8. **Available colors:** This is an array of multiple colors that you can set from the inspector.
9. **Random color:** This lets you choose whether select Available colors in order or just in a random fashion.

Visualizer Rotation Modifier

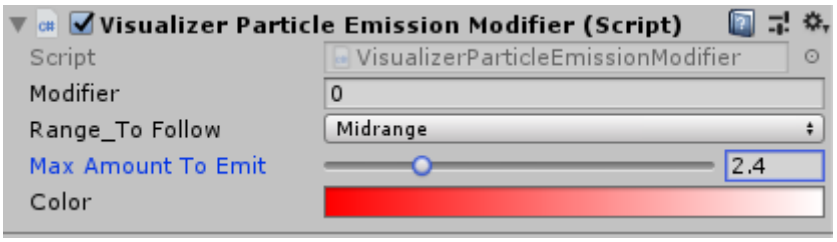
This component lets you rotate a game object according to the range to follow.



1. **Modifier:** This value will let you know in real time what's the intensity of the frequency range. This will update on real-time, no need to change it.
2. **Range to follow:** This dropdown will show all the frequency ranges available to follow.
3. **Multiplier:** This will let you choose with which intensity you want the object to change in each axis.
4. **Rotation Speed:** A multiplier for the rotation axes that determines the speed of the rotation.

Visualizer Particle Emission Modifier

This component will let you control the emission of a particle system according to the range to follow.



1. **Modifier:** This value will let you know in real time what's the intensity of the frequency range. This will update on real-time, no need to change it.
2. **Range to follow:** This dropdown will show all the frequency ranges available to follow.
3. **Max Amount to Emit:** This value will be multiplied by the modifier and will be the amount of particles the system will emit on each frame.
4. **Color:** This lets you choose a main gradient for the particle system.

Build your own modifier

In order to build your own modifiers you need to create a new script and make it inherit from **VisualizerObjectBase** and then call **EvaluateRange()** inside your Update() function , this will change the value of **Modifier** that you can use for whatever you need it.

This documentation is a work in progress and I'll be adding more guides and reference to the rest of the Modifiers. Feel free to extend the **VisualizerObjectBase** class to build your own.

Contact

Web: Kichex.itch.io
Forums: https://kichex.itch.io/apollovisualizer/community
Support email: Ekike797@gmail.com
Join our Discord server: Here