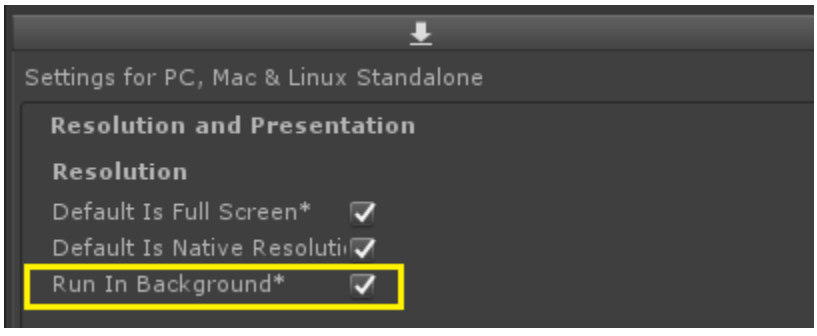
v2.2 DOCUMENTATION

FIRST STEPS

1. Make your application to run in background.
(Edit > Project Settings > Player > Resolution and Presentation >
Run in background)



2. Open the ExampleScene.

or...

1. Drag Rhythm Visualizator prefab to the scene.
2. Drag ExampleCamera to the scene.
3. Import a song and assign it to AudioSource script in Rhythm Visualizator Game Object.

VARIABLES DESCRIPTION

Restart Button: Reloads the **SoundBars** in **PlayMode**. Use when you change some options that need recreate the Sound Bars.

SCRIPT ASSIGNATIONS

Sound Bar Prefab From Center: The **SoundBarPrefab** used when **ScaleFrom is Center**.
Sound Bar Prefab From Downside: The **SoundBarPrefab** used when **ScaleFrom is Downside.**
Sound Bars Transform: SoundBars Center Transform.

AUDIO SETTINGS

Listen All Sounds: If **FALSE** it visualizes the assigned **AudioSource**, if **TRUE** visualize any listened sound in the scene.
AudioSource: The **AudioSource** to visualize if **Listen All Sounds** is **FALSE.**

SOUND BARS [Requires Restart]

Sound Bars Quantity: The quantity of Sound Bars to display. **Higher** values **decrease** the **performance.**
Scale From: The **Starting Scale Point** to **scale** the Sound Bars.
Sound Bars Width: The Width of the Sound Bars.
Sound Bars Depth: The Depth of the Sound Bars.

CAMERA

Center: The Center of the Sound Bars to Rotate Around.
Camera Control: If **TRUE** Rhythm Visualizator **controls** the **Main Camera** for you.
Rotate Camera: If **TRUE** your Camera will start **rotating around** the **Center Transform**.
Use Default Values: If **TRUE** when you change the **Visualization it uses the default values.** If **FALSE** it uses **below values.**
Velocity: Rotation Velocity. NEGATIVE VALUES: Rotate to Left. **POSITIVE VALUES**: Rotate to Right.
Height: The **Camera Height** View Point.
Orbit Distance: The **Orbit Distance** to the **Center.**

Field of View: Field of View of the Camera.

VISUALIZATION

Mirror: If **TRUE** the Sound Bars will be displayed in Mirror Mode. Bass is in the Center and scales equal for both sides.
Scale by Rhythm: The Sound Bars **scales only when the Rhythm are detected.**
Length: The Visualization Length.
Extra Scale Velocity: The **velocity to scale UP** the Sound Bars.

LEVELS

Global Scale: The Sound Bars Global Scale.
Min Height: The Sound Bars minimum height.
Smooth Velocity: The **smooth velocity to scale DOWN** the Sound Bars.
Channels: Large value of channels represents more spectrum values, you will need increase the SoundBars amount to represent all these values. **Recommended: 4096, 2048**
Method: FFTWindow to use, it is a type of filter. Rectangular = Very Low filter, BlackmanHarris = Very High filter. Recommended: Blackman

RHYTHM DETECTION [Experimental]

Rhythm Particle System: The Particle System to use when the Rhythm are detected.
Auto Rhythm Particles: Emit particles when Rhythm are detected.
Rhythm Sensibility: The Rhythm detection sensibility.
Amount to Emit: Amount of Particles to emit when the Rhythm are detected.
Rhythm Particles Max Interval: The max interval in seconds to emit the particles.

BASS / MIRROR SETTINGS Mirror Mode uses these settings

Bass Sensibility: Visualization Bass Sensibility. **Higher values increases the scale.**
Bass Height: Increases the **height visualization of the Bass.**
Bass Horizontal Scale: Higher values display less bass spectrum values.
Bass Offset: Apply offset to the spectrum values for display or not some bass frequencys.

TREBLE SETTINGS

Treble Sensibility: Visualization Treble Sensibility. **Higher values increases the scale.**
Treble Height: Increases the height visualization of the treble.
Treble Horizontal Scale: Higher values display less treble spectrum values.
Treble Offset: Apply offset to the spectrum values for display or not some treble frequencys.

APPEARANCE

Sound Bars Particles: if **TRUE** every Sound Bar will emit particles.
Particles Max Interval: The Interval to Emit Particles. **Higher values emit less particles but increase the performance.**
Min Particle Sensibility: The minimum value to emit a particle.
Lerp Color: If **TRUE** Lerps the visualization colors using the below colors.
Colors: Colors to use when Lerp Color is **TRUE**.
Color Interval Time: Time in seconds to change from one color to other.
Color Lerp Time: Time in seconds to lerp one color to other.
Use Gradient: if **TRUE** uses the below gradient in the visualization.

Gradient: The gradient to use in the visualization.
Rhythm Particles System Color: The Rhythm Particle System Color to emit when **Use Gradient** is **TRUE**.

RAYS [Requires Restart]

Rays Length: The Rays Length.
Rays Alpha: The Alpha for the Rays.

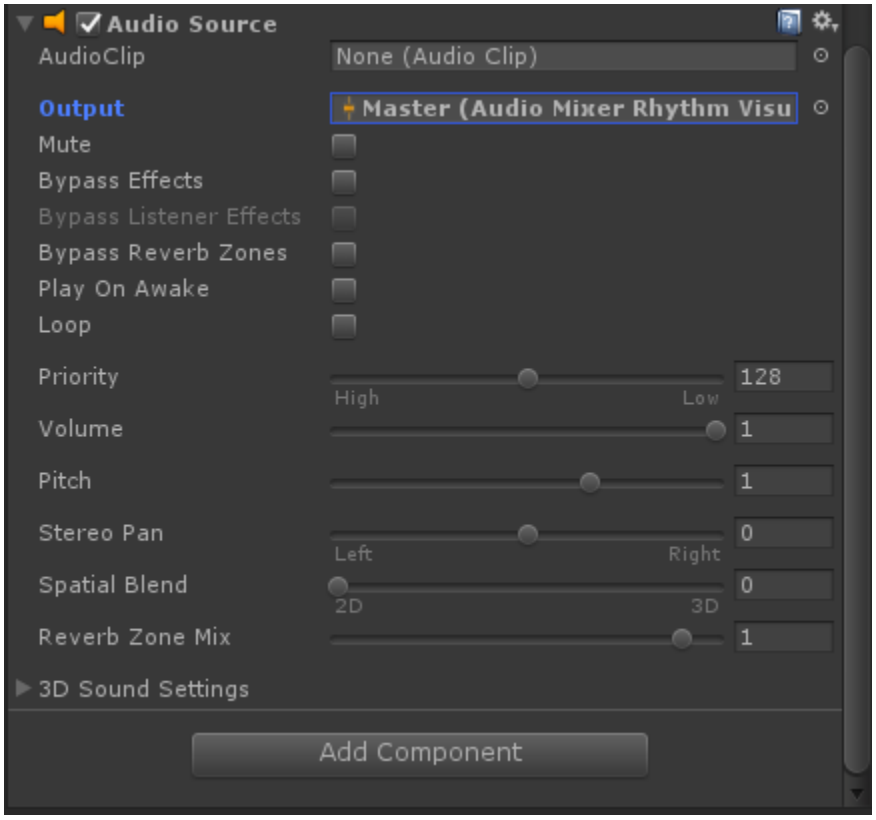
RHYTHM VISUALIZATOR FAQ

THE AUDIO MIXER CONTROLS THE VOLUME AND THE AUDIOSOURCE CONTROLS THE INPUT TO THE VISUALIZATION

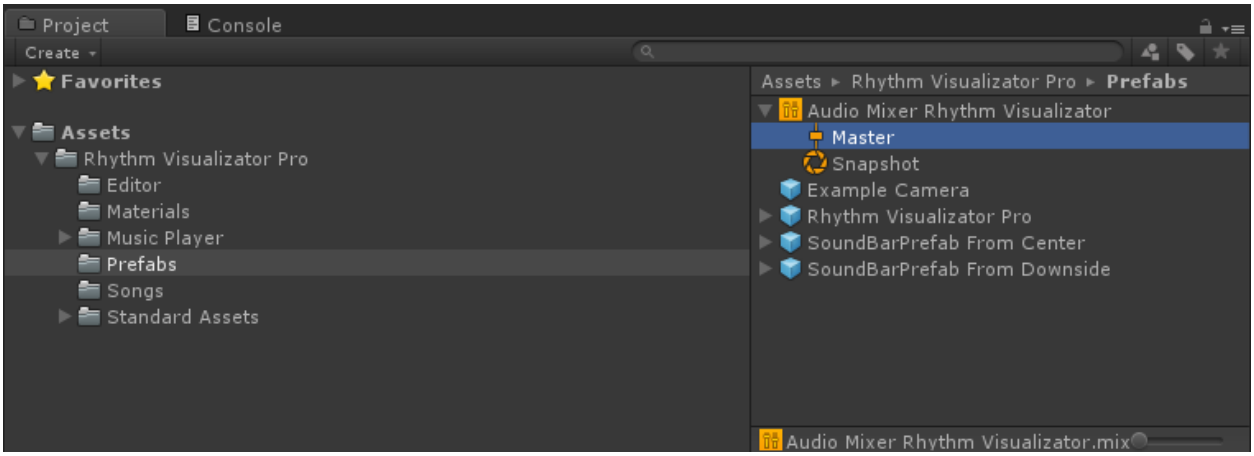
LEAVE THE AUDIOSOURCE UNMUTED AND SET HIS VOLUME TO 1

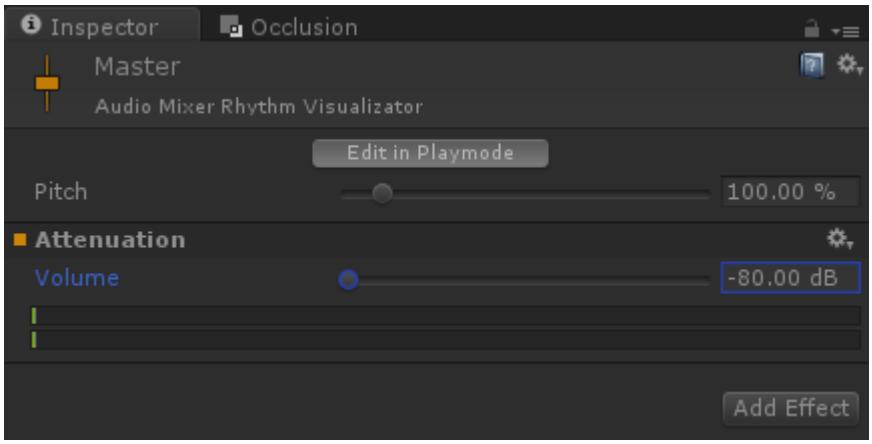
HOW TO VISUALIZE AN AUDIOSOURCE IF THE VOLUME IS ZERO OR MUTED?

1. Set AudioSource Volume to 1 and unmark Mute.
2. Make sure the AudioSource have the Master AudioMixer.



3. Search your AudioMixer and open “Master”.



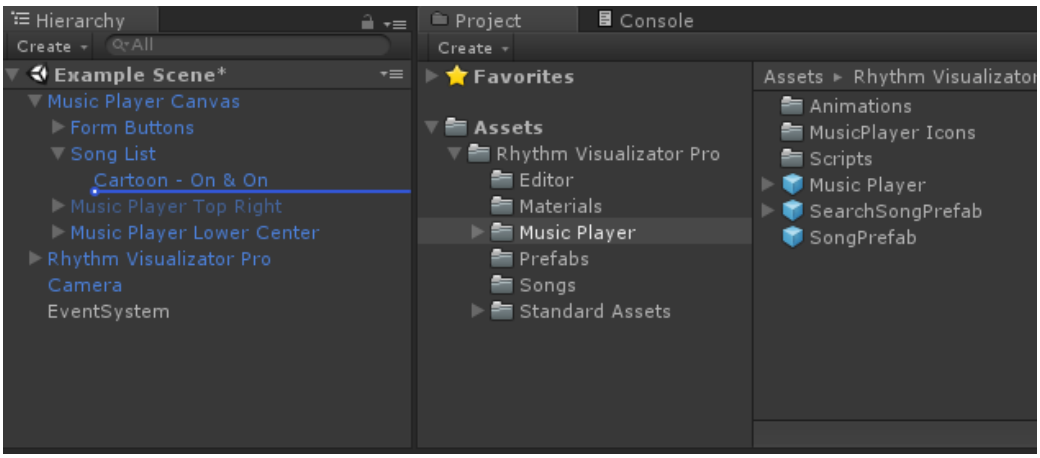


4. Reduce the volume in the AudioMixer (MUTED: -80.00dB - NORMAL: 0.00dB)

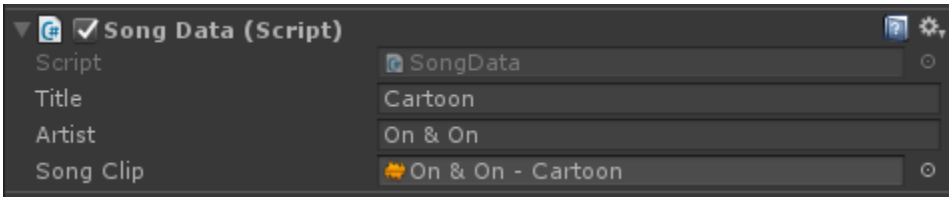
HOW TO ADD A NEW SONG?

You really do not need to use the Music Player to listen songs; You can use Rhythm Visualizator without using it

1. Make sure you have "Music Player" prefab in the Scene.
2. Open "Song List" transform.
3. Drag and Drop the "SongPrefab"to "SongList" transform.



4. Assign "Artist" and "Title" values and his "AudioClip"on "SongData"script.



Thanks for purchasing!

Created by Carlos Arturo Rodríguez Silva

Contact:

carlosarturors@gmail.com

Video v2:
https://www.youtube.com/watch?v=72X0OySMzj0

Video v1 (old):
https://www.youtube.com/watch?v=LXYWPNltY0s

CHANGES

---

v2.2

Added

- Microphone support
- Microphone example scene

Changed:

- Added to the Documentation a clarification for Audio Mixers and Audio Sources

v2.1

Added

- Mirror Mode (I love this)
- Square Visualization
- Post Processing Stack Implementation
- Sound Bars Depth

Changed:

- Default values (Length, Rhythm Sensibility, Treble and Bass Settings, Gradient Color, Rays Length, Channels)
- Mirror is set to true by default
- Headers
- Particles duration changed from 4 to 8 sec
- Some variables has been renamed

Fixed/Deleted:

- The SoundBars variable is now private and can be restarted without error
- Standard Assets deleted (Except Water)
- Legacy Camera Effects has been deleted
- Example scene v1 deleted
- ToolTips (added to Documentation)

v2.0

Added

- Customizable Light Rays
- Customizable Gradient Color
- Now you can control the Minimum SoundBars Height
- Now you can control the SoundBars Width
- Restart Button, you don't need to Stop and Play Unity when you change some values
- New Example Scene "Rays Only"
- New Example Scene "No Reflections"
- New Example Scene "Sun Light"
- New Original Example Scene
- SoundBarTransform added as variable

Changed:

- Downside Music Player has been changed
- Some default values have been changed
- All example scenes have been changed
- Spectrum Detection has been improved
- Rhythm Visualizator doesn't need the SoundBar tag anymore

v1.2

Added

- New example scene “Muted Audio”
- New variable “Use Default Camera On Change”this will use the predefined values for the camera when you change the visualization
- Another Music Player
- Added next and previous visualization texts to the default scene
- Added more prefabs

Changed:

- Default Rhythm Sensibility value changed to 30
- Some controls order has been changed
- Default colors has been changed
- “Example Scene No Music Player” has been changed and now includes First Person Controller

Fixed:

- Various Music Players fails on certains situations, now you can use all you want and it will works good
- Scale By Rhythm variable works again

v1.1

Added

- Now you can set any quantity of colors that you want
- New “Height” variable to set the height of the camera
- Listen All Sounds Button, now with one clic, the visualizator will hear all sounds
- New Example Scene No Music Player, here will show how Rhythm Visualizator does not need the Music Player to work.
- Added “EmitIfThereAreRhythm”function, if you call it they will analyze at the exact moment if there are Rhythm and emits particles.
- The two music players are displayed at the same time in the example scene

Fixed:

- Fixed some problems if you rotate the visualizator
- Rhythm Detection improved
- Forms variable renamed to “Visualizations”

v1.0 First Version