

Waveform

Audio waveform drawing utility for Unity

Overview	2
What is Waveform	2
Package Contents	2
Getting Started	2
Adding a Waveform	2
Customization	4
Waveform Display	4
Properties	5
Read Only Properties	6
About / Contact	6

Overview

What is Waveform

Waveform is an extension that allows you to draw the waveform of an AudioClip asset inside your own editor windows or inspectors. Waveforms will display within scroll views (if necessary) and will provide you, by means of a callback, to draw your own content on-top of the waveform and within the scrollview. This makes Waveform the perfect extension to kick start the editor for that rhythm game you wanted to make, audio basic event trigger system, or any other tool you can think of that would benefit from the rendering of an audio waveform.

Package Contents

Once you've downloaded and imported the package from the asset store you should have a new menu item "Tools / Waveform / Example". This example window provides you with a basic example of how you can integrate a WaveformDrawer into an editor window, all you have to do is open up the window and add an AudioClip asset to the "Audio" field provided. This example window also lets you experiment with the various configurable settings that are covered later in this document for customizing the look of the waveform. Additionally there is a test object (scriptableObject) with a custom inspector showing how a waveform could be rendered into an inspector window instead of a custom editor window.

Getting Started

Adding a Waveform

Adding a waveform to your own editors is simple, and involves the following steps:

1. Declare a WaveformDrawer variable in your editor.
2. Create your WaveformDrawer and give it an AudioClip (or null if you don't have one yet).
3. Give it an audio clip (if not already provided).
4. Call the Draw method of the WaveformDrawer specifying the desired width, height, and optional callback for custom overall.

The following script example is a bare bones window for displaying a waveform, with all the other clutter removed.

```

using UnityEngine;
using UnityEditor;
using Waveform;

public class BasicWaveform : EditorWindow
{
    private AudioClip audioClip;
    private WaveformDrawer waveform;

    [MenuItem( "Tools/Basic Waveform" )]
    private static void ShowWindow()
    {
        EditorWindow.GetWindow<BasicWaveform>();
    }

    private void OnGUI()
    {
        // Create the WaveformDrawer if we haven't yet done so.
        if(waveform == null)
        {
            waveform = new WaveformDrawer( audioClip );
        }

        // Specify an audio clip, as we need something to draw.
        AudioClip newClip = EditorGUILayout.ObjectField( audioClip, typeof( AudioClip ), false ) as AudioClip;

        // If it changes or the waveform hasn't got it yet (happens on assembly reload) provide the new clip.
        if( newClip != audioClip || waveform.AudioClip == null )
        {
            audioClip = newClip;
            waveform.SetClip( audioClip );
        }

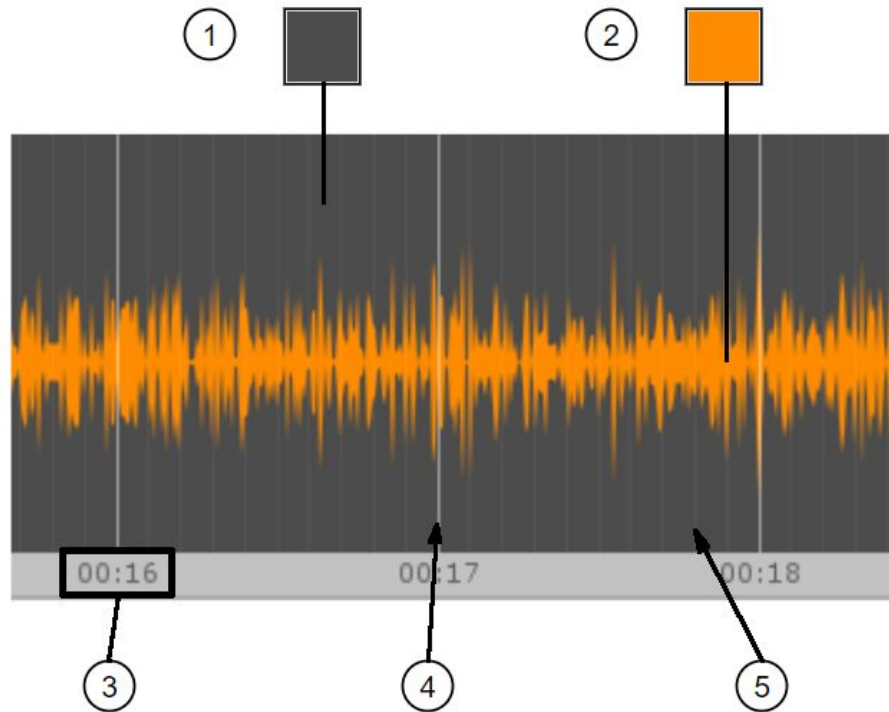
        // Then heres the bit that actually actually calls the draw for the audio waveform.
        // We'll draw it the full width of the editor window, with a height of 200.
        waveform.Draw( CustomOverlay, position.width, 250 );
    }

    private void CustomOverlay( Rect rect, Vector2 scrollPos, float widthPerSecond )
    {
        // Do some custom stuff here!
    }
}

```

Customization

Waveform Display



(Forgive the terrible image work here, this is what happens when you finish your documentation at 2am)

1	Background Colour
2	Waveform Colour
3	Timestamps
4	Timelines
5	Subdivided Timelines

Properties

In order to allow Waveform to fit your needs, spice things up, or just match the rest of your custom tool, there are several configurable properties in a WaveformDrawer. Each of these properties can be set after constructing your waveform object and can be changed at any time during the waveforms lifetime.

Property	Comments
HorizontalTracks	Integer value that controls the number of alternating dark/light bands in the background of the waveform draw.
CurveColour	Colour value for the waveform itself.
BackgroundColour	Colour value for the background the waveform is drawn on.
TimelinesColour	Colour value for of the vertical lines denoting time along the waveform.
DrawTimeLines	True if you want vertical lines at one second intervals along the waveform, False to hide these lines.
SubdivideTimeLines	True if you want each of those one second increments to be subdivided by fainter lines at each tenth of a second. False to hide these lines.
DrawTimestamps	True to display the time along the bottom of the waveform, increments will automatically adjust based on scale. False to hide these labels.
MinWidthPerSecond	Minimum width in GUI space for one second of the waveform to cover. Effectively controls the display size and minimum zoom.
MaxWidthPerSecond	Maximum width in GUI space for one second of the waveform to cover. Effectively controls the display size and maximum zoom.
Zoom	0-1 value to allowing zooming in/out on the waveform.

Read Only Properties

Additionally there are some properties that provide access to certain values that you may wish to access in the containing editor code. However these values cannot be directly set.

Property	Comments
ScrollPos	Vector2 representing the current scroll position of the WaveformDrawers scrollview region.
WidthPerSecond	Current width per second that the waveform is rendering at in GUI space based on current min, max, and zoom settings.
AudioClip	Current audio clip that this waveform is drawing.

About / Contact

If you have any issues with the use of this extension, have a feature request, bug report, or general feedback, please do not hesitate to get in touch with me at the email address listed below. Please note that these assets are developed outside my normal day job and so whilst I will try to respond to all raised issues as soon as possible, there may be some delays, I apologise for any inconvenience this may cause.

Email: assets@fatwednesday.co.uk