VIVALDI

Operation Manual

Content

1 Basics 4
1.1 Create a Vivaldi Composer 4
1.2 Create a new Composition 5
1.3 Open the Vivaldi Editor 5
2 Editor 6
2.1 Cue Points 6
2.1.1 Add a new Cue Point 6
2.1.2 Remove a Cue Point 6
2.2 Loops 7
2.2.1 Add a new Loop 8
2.2.2 Remove a Loop 8
2.3 Tracks 9
2.3.1 Add a new track 9
2.3.2 Remove a track 9
2.3.3 Move a single Track 10
2.3.4 Move multiple Tracks 10
2.3.5 Duplicate a Track 11
2.3.6 Crop a Track 11
3 Runtime 12
3.1 Pool 12
3.2 Scripting 13
3.2.1 Play a Composition 13
3.2.2 Registering events 14
4 Anchors 15
5 Composition Player 16
6 Zone Player 17
7 Track Settings 18
7.1 General 19
7.2 3D Settings 20

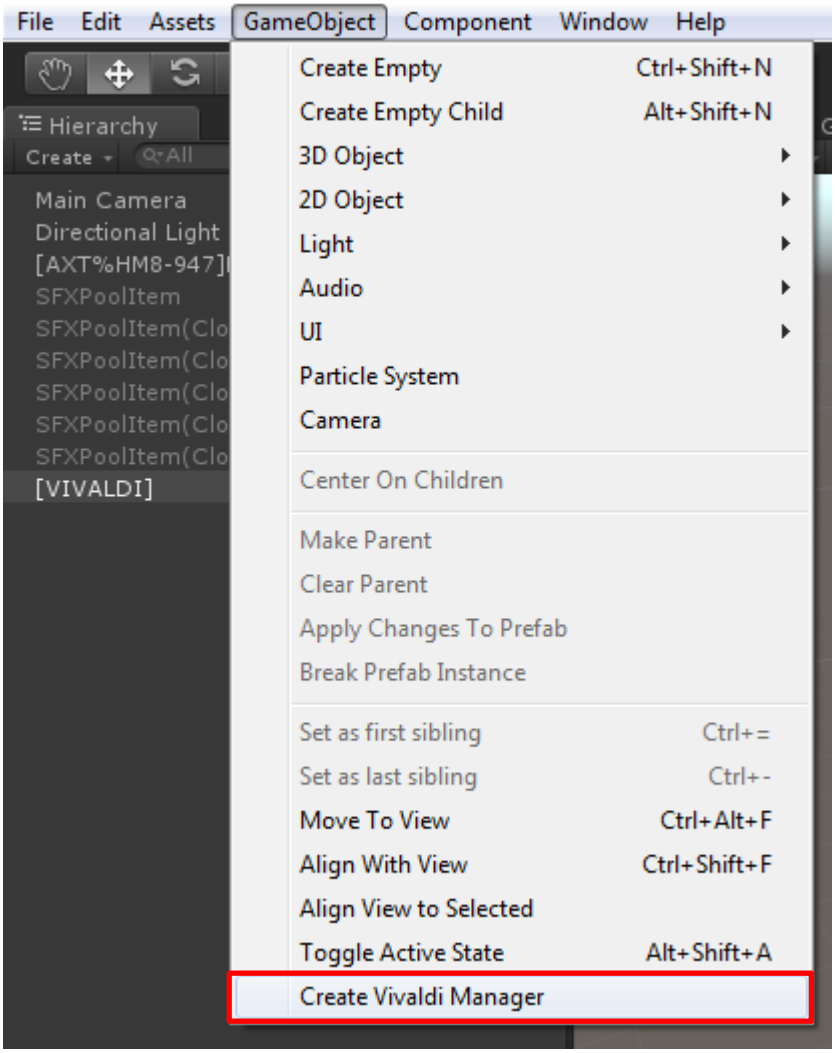
1 Basics

1.1 Create a Vivaldi Composer

Every project making use of the Vivaldi system needs to have a `VivaldiComposer` object.

The `VivaldiComposer` object is responsible for playing and pooling the different compositions and their containing sound effects.

In the main menu go to **GameObject > Audio >Create Vivaldi Manager** to add the manager to your current scene.



Note that the `VivaldiComposer` object is a singleton and doesn't needs to be created in every scene. Just create it on your entry scene.

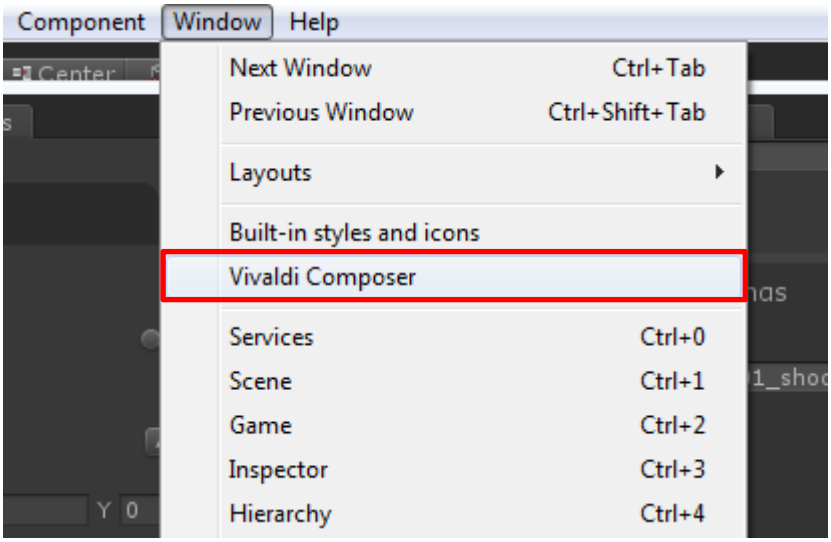
1.2 Create a new Composition

1. Somewhere in your project view right-click to open the context menu.
2. Click **Create > Vivaldi > Composition**

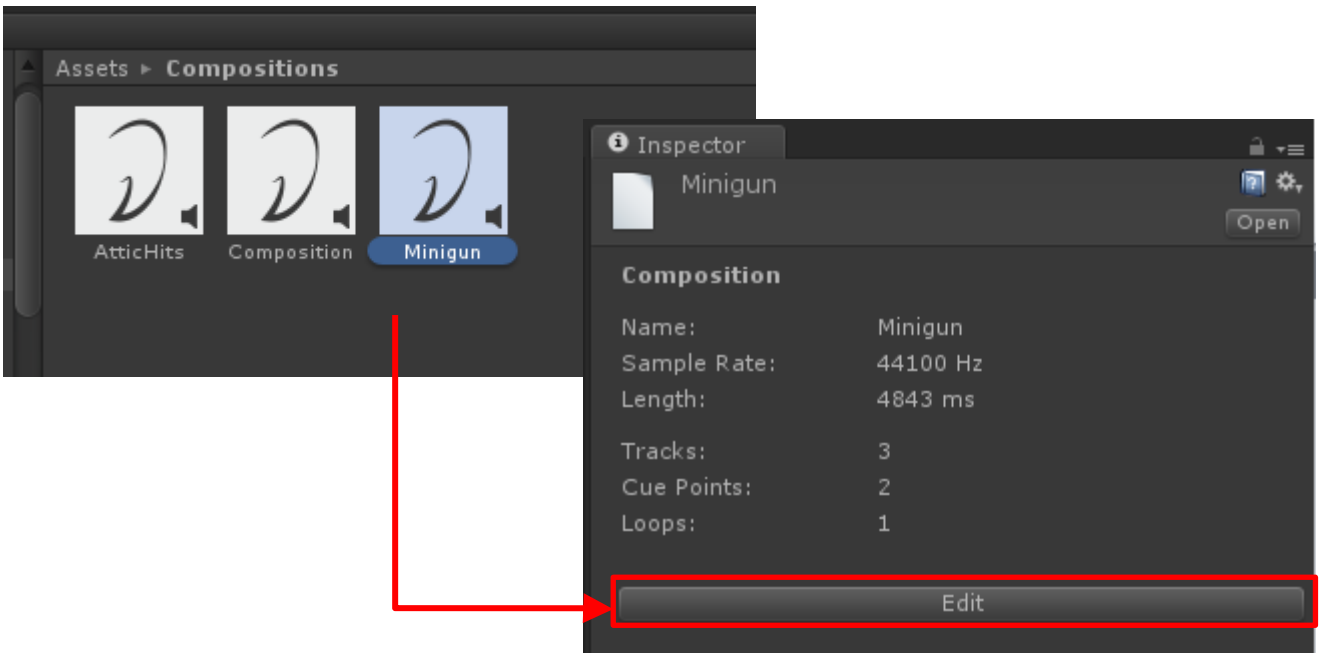
Alternatively, in the main menu go to **Assets > Create > Vivaldi > Composition.**

1.3 Open the Vivaldi Editor

In the main menu go to **Window > Vivaldi Composer** to open the Vivaldi editor window.



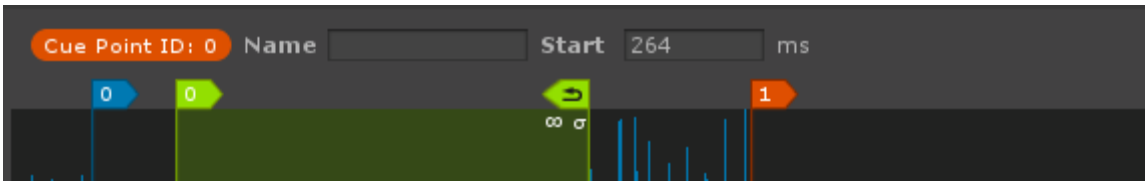
Alternatively, click to an already created composition asset and click the **Edit** button in the inspector**.**



2 Editor

2.1 Cue Points

Cue points are user-defined time locations in a composition. Those points throw events when getting reached in the timeline and can be used as jump targets by the API.



Cue points consist of a unique ID, a start position in milliseconds and an optional name.

Parameter Name	Description
Cue Point ID	The unique Identifier for this cue point.
Name	An optional name for the cue point.
Start	The start of the cue point in milliseconds.

2.1.1 Add a new Cue Point

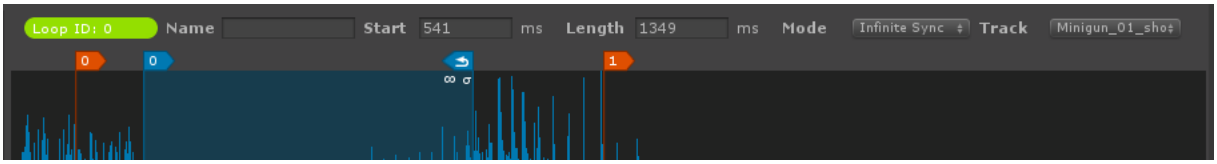
1. Open the context menu by right-clicking somewhere on the global waveform preview.
2. Select the **Add Cue Point** command in the context menu.
3. The cue point can be moved by dragging its flag head.

2.1.2 Remove a Cue Point

An active (blue) cue point can be removed from the current composition by pressing **Delete** or **Backspace**.

2.2 Loops

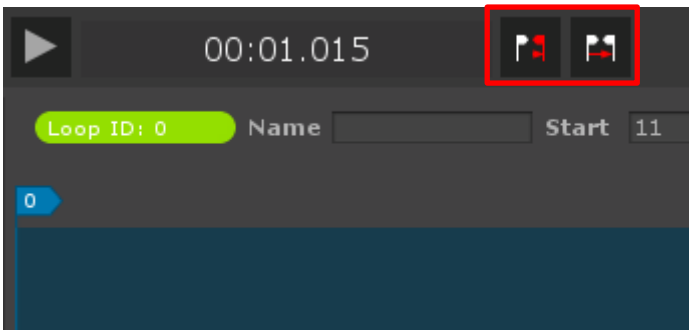
Loops are user-defined timespans in a composition. Loops throw events when getting reached, passed and left in the timeline.



Parameter Name	Description
Loop ID	The unique Identifier for this loop.
Name	An optional name for the loop.
Start	The start of the loop in milliseconds.
Length	The length of the loop in milliseconds.
Mode	**Fixed** Plays the loop for a fixed amount of times. **Infinite** Plays the loop an infinite amount of times until the loop gets manually left. **Fixed Sync** Plays the loop for a fixed amount of times. The loop gets synced to a track's samples. Use this for seamlessly looping over a track. **Infinite Sync** Plays the loop an infinite amount of times until the loop gets manually left. The loop gets synced to a track's samples. Use this for seamlessly looping over a track.
Track	Sets the track used in syncing mode.

Note that unsynchronized loops are not necessarily seamless as the accuracy of the start of a loop depends on the current framerate.

In the editor where you don't have access to the Vivaldi API, a loop can be left by the buttons seen in the image below.



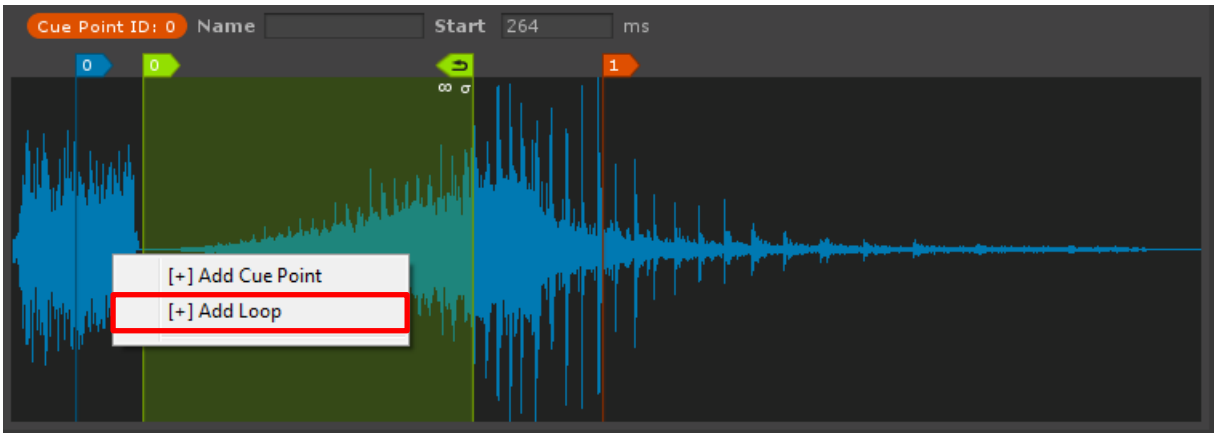
Hard Leave
Immediate jumps out of the active loop.

Soft Leave
Finishes the current iteration before leaving.

2.2.1 Add a new Loop

1. Open the context menu by right-clicking somewhere on the global waveform preview.
2. Select the **Add Loop** command in the context menu.
3. The loop can be moved by dragging its flag head or its body.

Hold down the **SHIFT** key to snap to tracks, cue points or loops.



2.2.2 Remove a Loop

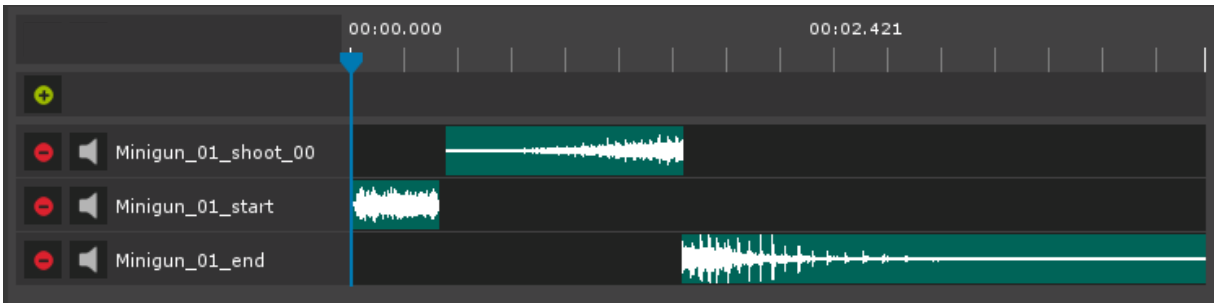
An active (blue) loop can be removed from the current composition by pressing **Delete** or **Backspace**.

Alternatively, you could delete it by its context menu (right-click).

Note that your mouse point needs to be located in the preview area to delete a loop point with your keyboard.

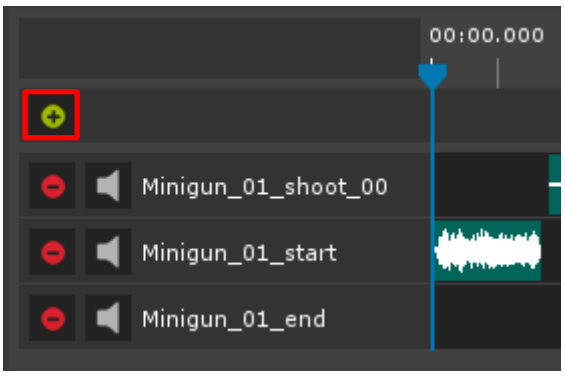
2.3 Tracks

A track represents a single audio clip in the composition's timeline.



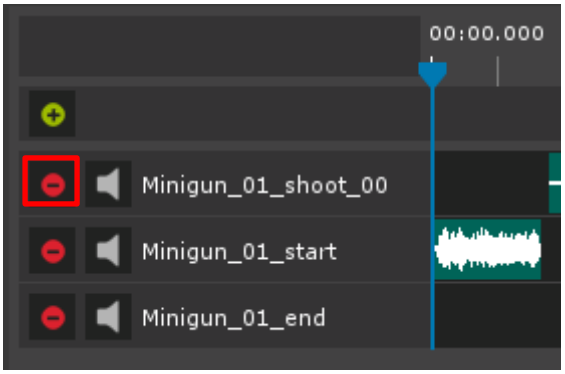
2.3.1 Add a new track

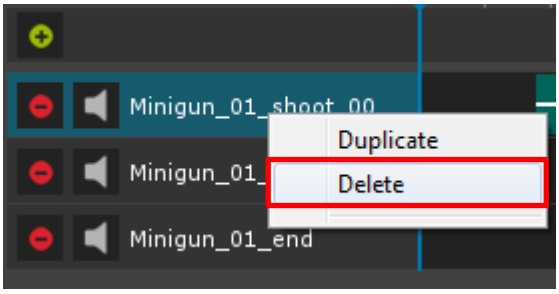
Add a new track by clicking on the green plus button.



2.3.2 Remove a track

Remove a track by clicking on the red minus button or via its corresponding context menu opened by a right-click.



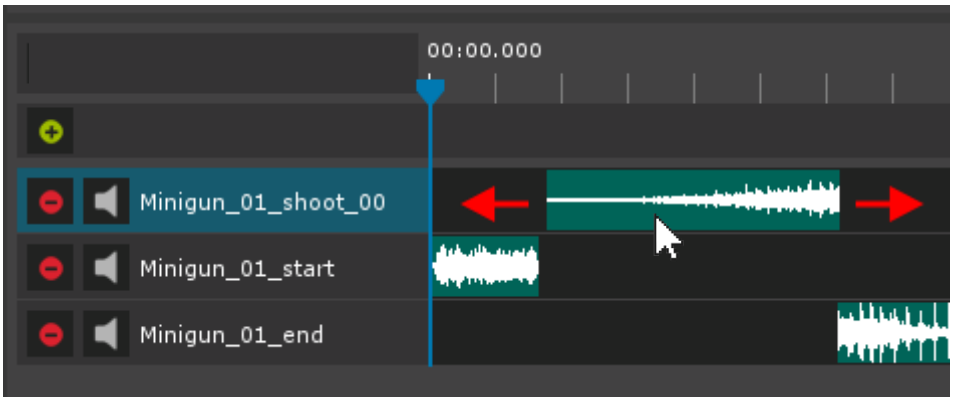


2.3.3 Move a single Track

Move a track by dragging it in the sequencer.

Hold down the **ALT** key to snap the track at a 100ms grid.

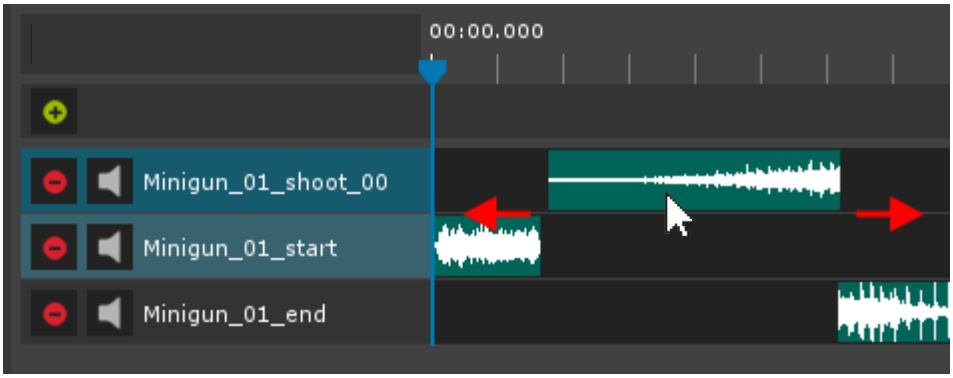
Hold down the **SHIFT** key to snap to other tracks, cue points or loops.



2.3.4 Move multiple Tracks

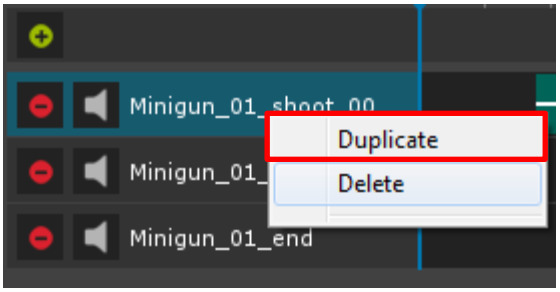
Select all tracks that you want to move with the **SHIFT** key held down and drag the tracks in the sequencer.

Hold down the **ALT** key to snap the tracks at a 100ms grid.



2.3.5 Duplicate a Track

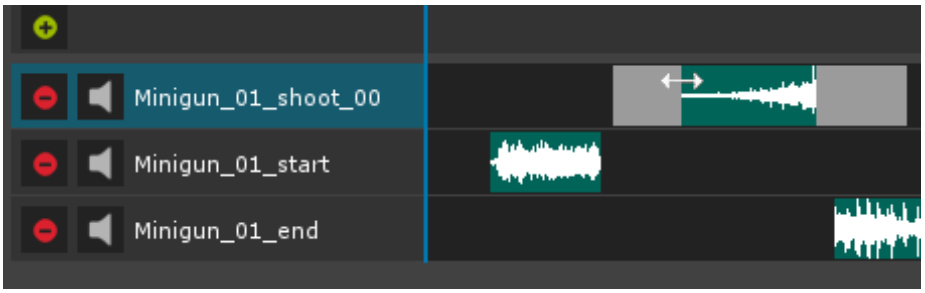
Right-click to the track that you want to be duplicated and click **Duplicate**.



2.3.6 Crop a Track

Sometimes you don't want to store multiple versions of the same audio file just for being able to play different parts out of it.

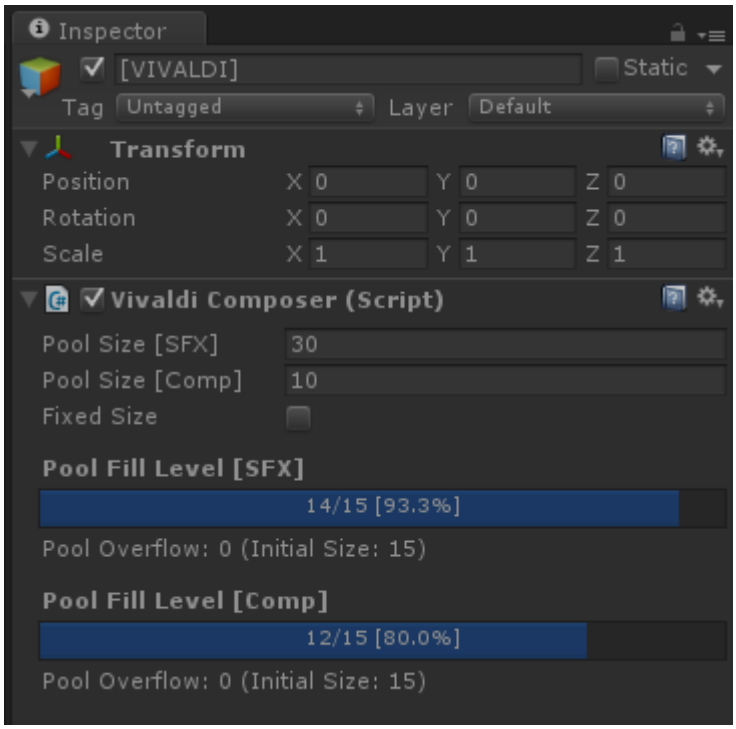
Crop a track by dragging one of its borders toward the center of the track to change its playing region. The grayed-out regions will not be played.



3 Runtime

3.1 Pool

Compositions and single tracks get pooled independently from each other. This ensures that a single composition does not occupy pool space for a track that is not looped and has already been finished.

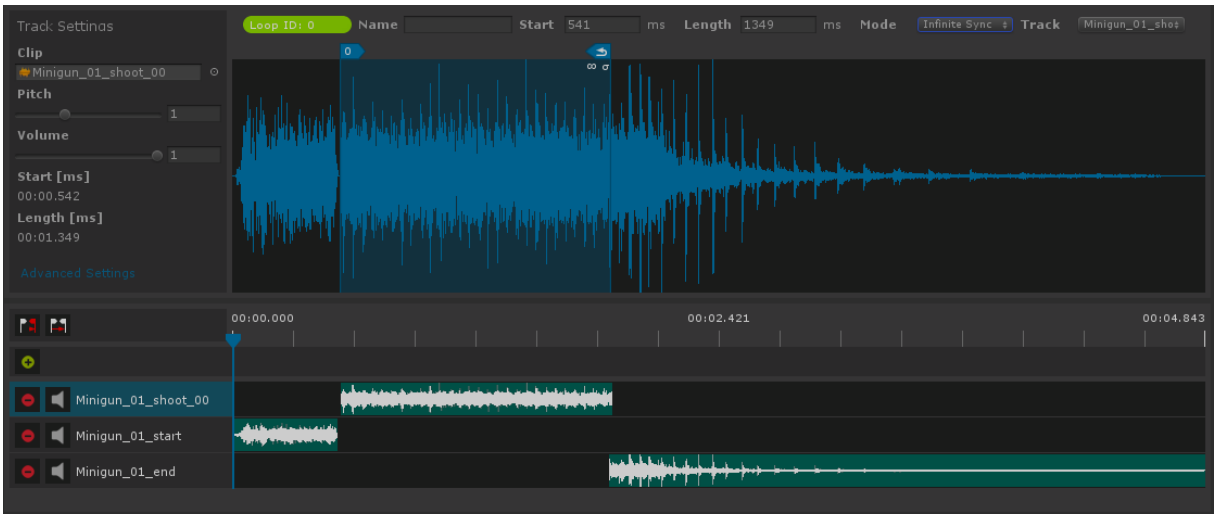


Parameter Name	Description
Pool Size [SFX]	The initial pool size for the single audio clips.
Pool Size [Comp]	The initial pool size for the composition.
Fixed Size	When disabled the pool is allowed to dynamically grow when needed.
Pool Fill Level [SFX]	Visualizes the current fill level of the SFX pool.
Pool Fill Level [Comp]	Visualizes the current fill level of the composition pool.

3.2 Scripting

3.2.1 Play a Composition

The following snippet shows how to implement a simple gun sound system that plays a loop while holding down the virtual Shoot button.




```
using ch.sycoforge.Vivaldi;
using UnityEngine;

public class SFXDemo : MonoBehaviour
{
    public VivaldiComposition MinigunSFX;
    private CompositionControl control;


    void Update()
    {
        if (Input.GetButtonDown("Shoot"))
        {
            control = MinigunSFX.Play();
            control.OnCuePointReached += control_OnCuePointReached;
            control.OnStarted += control_OnStarted;
            control.OnFinished += control_OnFinished;
        }

        if (Input.GetButtonUp("Shoot"))
        {
            // Check if control handle has not been collected by the pool
            if (control.IsValid)
            {
                // Immediate leave all the actives loops
                control.LeaveActiveLoops(true);
            }
        }
    }
}
```

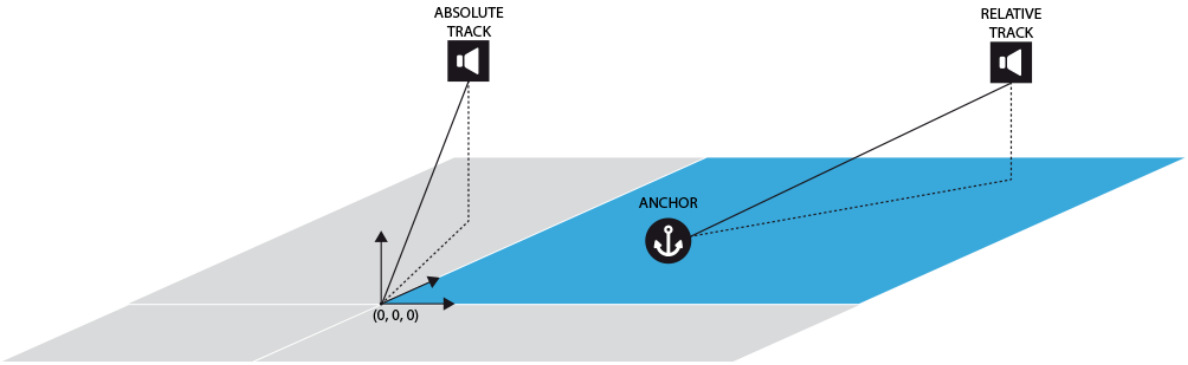
3.2.2 Registering events

The following snippet shows how to hook events to a runtime composition.

```
using ch.sycoforge.Vivaldi;
using UnityEngine;

public class SFXDemo : MonoBehaviour
{
    public VivaldiComposition MinigunSFX;
    private CompositionControl control;


    void Update()
    {
        if (Input.GetButtonDown("Shoot"))
        {
            control = MinigunSFX.Play();
            control.OnCuePointReached += control_OnCuePointReached;
            control.OnStarted += control_OnStarted;
            control.OnFinished += control_OnFinished;
        }

        if (Input.GetButtonUp("Shoot"))
        {
            // Check if control handle has not been collected by the pool
            if (control.IsValid)
            {
                // Jump to the first cue point
                control.GoTo(control.Composition.CuePoints[0]);
            }
        }
    }

    void control_OnCuePointReached(CompositionControl composition, CuePoint cuePoint)
    {
        Debug.Log("CuePointReached: " + cuePoint.ID);
    }

    void control_OnStarted(CompositionControl composition)
    {
        Debug.Log("Composition OnStarted: " + composition);
    }

    void control_OnFinished(CompositionControl composition)
    {
        Debug.Log("Composition OnFinished: " + composition);
    }
}
```

4 Anchors

A composition can be anchored to a `Transform` component of a `GameObject`. This allows a relative positioning of each track.

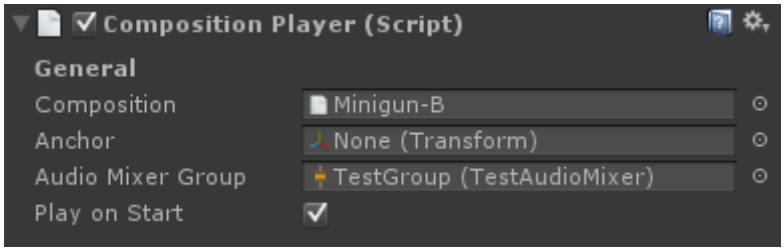


The following snippet shows how to anchor a composition to a specified `GameObject`.

```
using ch.sycoforge.Vivaldi;
using UnityEngine;

public class SFXDemo : MonoBehaviour
{
    public VivaldiComposition MinigunSFX;
    public GameObject AnchorObject;
    private CompositionControl control;


    void Update()
    {
        if (Input.GetButtonDown("Shoot"))
        {
            control = MinigunSFX.Play(AnchorObject.transform);
        }

        if (Input.GetButtonUp("Shoot"))
        {
            // Check if control handle has not been collected by the pool
            if (control.IsValid)
            {
                // Jump to the first cue point
                control.GoTo(control.Composition.CuePoints[0]);
            }
        }
    }
}
```

5 Composition Player

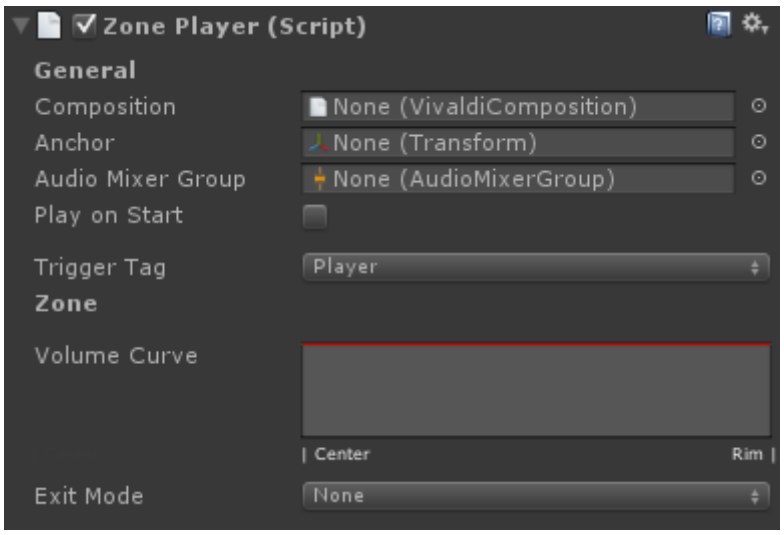
The `CompositionPlayer` component can be used to manage a composition. The component exposes basic methods to play and stop a composition.



Parameter Name	Description
Composition	The composition controlled by the player.
Anchor	The transform used for anchoring the composition.
Audio Mixer Group	The audio mixer group used to route the audio data to.
Play on Start	Specifies whether the composition should be play when starting up.

6 Zone Player

The `ZonePlayer` component can be used to manage a composition within a certain zone (volume). The component exposes the same methods as the `CompositionPlayer` component, but extends its functionality by triggers and volume curves.

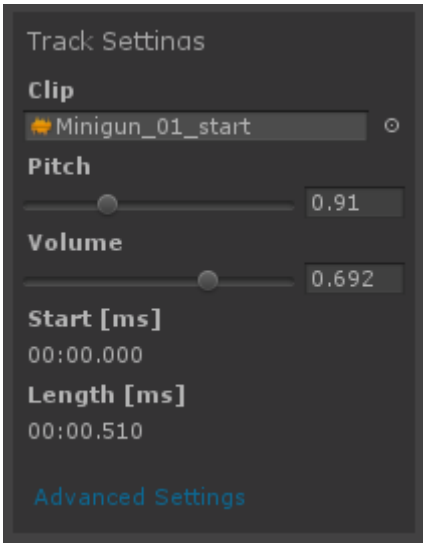


Parameter Name	Description
Composition	The composition controlled by the player.
Anchor	The transform used for anchoring the composition.
Audio Mixer Group	The audio mixer group used to route the audio data to.
Play on Start	Specifies whether the composition should be played when starting up.
Volume Curve	Defines a normalized curve describing the volume from the center of the zone to one of its borders.
Exit Mode	The action that happens when leaving the Collider volume. **None**: Does nothing when exiting. **Stop**: Immediately stops the composition. **Fade Out**: Linearly fades out the composition.

The `ZonePlayer` component needs a `Collider` to detect collisions.

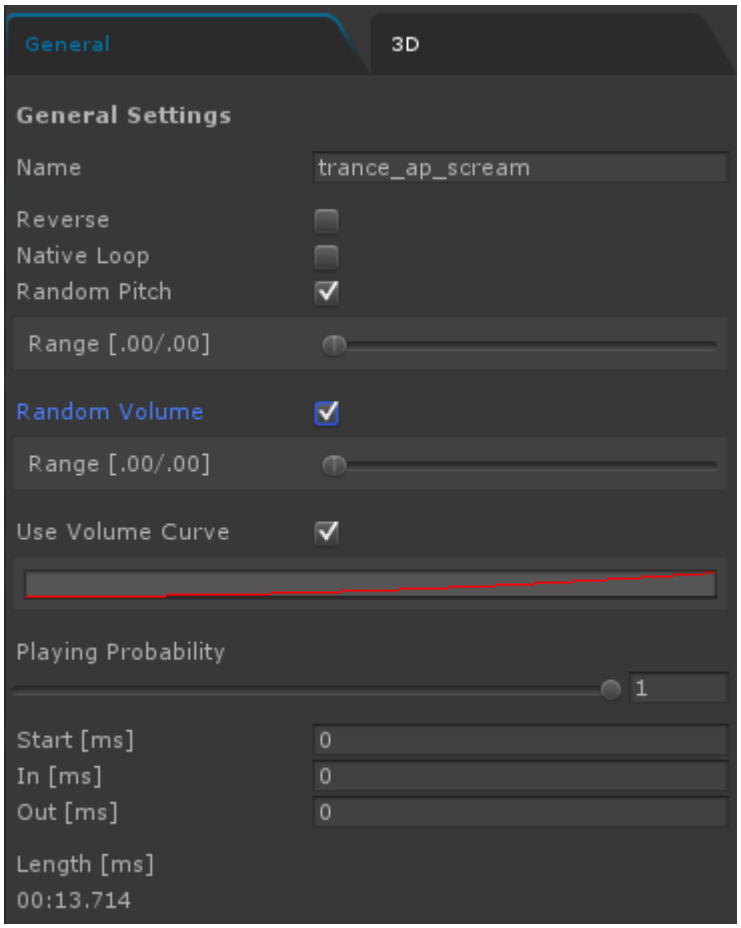
7 Track Settings

When selecting a track in the sequencer the corresponding properties can be changed in the dialog to the left of the waveform preview.



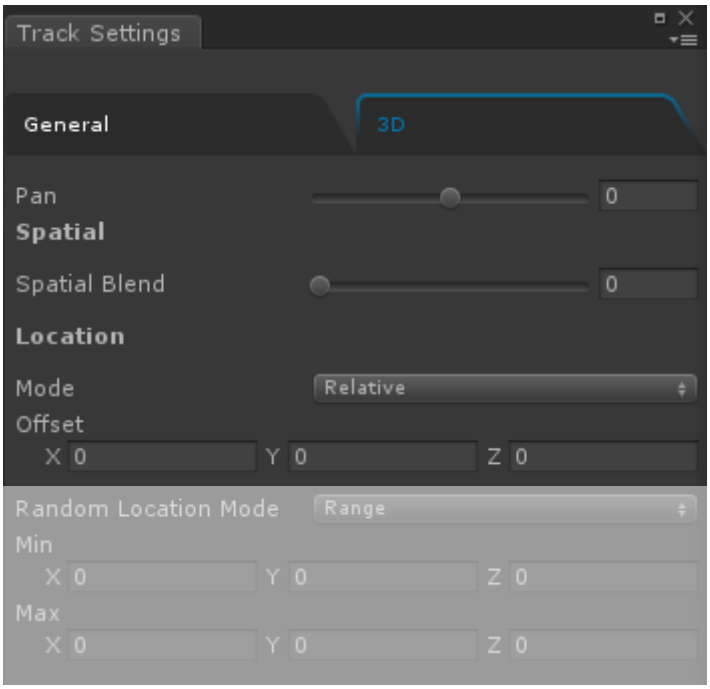
Parameter Name	Description
Clip	The `AudioClip` of the track.
Pitch	The static pitch shift of the track.
Volume	The static volume of the track.
Start [Read Only]	The start position of the track within the composition in milliseconds.
Length [Read Only]	The length of the track in milliseconds.

7.1 General

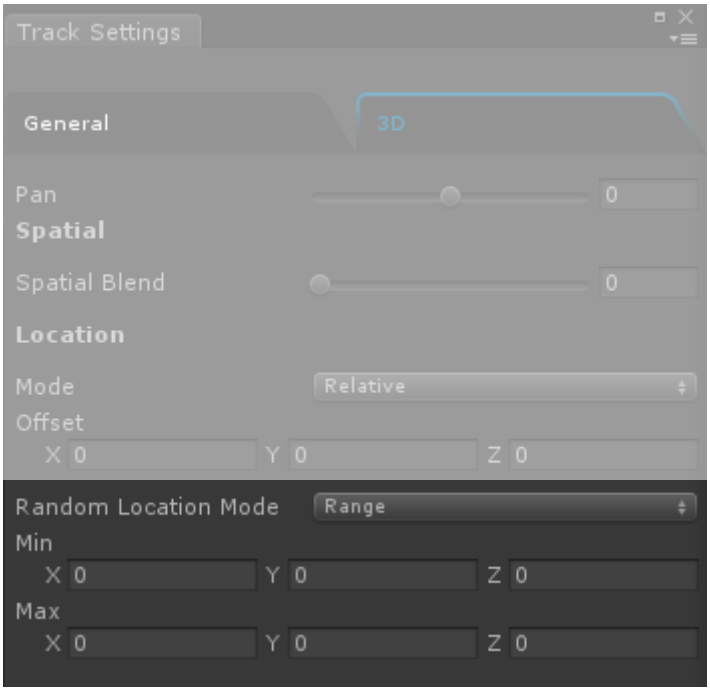


Parameter Name	Description
Name	The tracks name.
Reverse	Changes the playing direction of the track.
Force Loop	Loops the track regardless if it's part of a loop or not. Note that the track gets looped over the complete composition and never gets collected by the pooling system as long as the composition is alive.
Random Pitch	Sets a random range for the pitch shift.
Random Volume	Sets a random range for the volume.
Use Volume Curve	Sets a curve used for sampling the track volume. The curve value gets multiplied by either the random or the standard volume.
Playing Probability	Sets the chance that the track gets played. When set to 1, the track gets played every time. When set to 0.5 the track gets played with a probability of 50%.

7.2 3D Settings



Parameter Name	Description
Pan	The stereo pan of the track [-1..1]. A value of -1 means completely left and 1 completely right.
Spatial Blend	Blends the track between 2D and 3D. When set to 0 the track gets played completely in 2D. When set to 1 the track gets played completely in 3D. Note: The location settings only get accessible when Spatial Blend > 0.
Mode	Sets the location mode. **Relative** The track gets played relative to the anchor specified in the API's `Play(..)` method. **Absolute** The track gets played at the specified location, ignoring the supplied anchor.
Offset/Location	**Offset** The vector used to offset the track relative to the anchor. **Location** The absolute location to play the track at.



Parameter Name	Description
Random Location Mode	The mode used for the random offset added to the track's position in 3D space. **None** No random offsetting used. **Range** Adding a random offset specified by a minimum and maximum value. **Spherical** Adding a random offset specified by a sphere located at the anchor point.
Min/Max	The range used to offset the track's position. (Range Mode)
Max Radius	The maximum radius used to offset the track's position. (Spherical Mode)