Tazman-Audio

Fabric Manual

FABRIC

Table of contents

Introduction

Fabric extends Unity's existing audio functionality and provides an extensive set of high level audio components that allows the creation of complex and rich audio behaviours.

Key benefits of Fabric are:

- **It allows the design of many types of audio behaviours quickly and easily.**
- **All game audio assets are located under one hierarchy making it easy to manage and locate.**
- **Quick integration with the game using an easy and intuitive event base system. Trigger audio in the editor or from code.**
- **Custom user interfaces allow to quickly iterate and focus on the audio functionality that is important.**
- **Multiple instances of the same audio behaviour can be triggered from different Game Objects.**
- **Written entirely with Unity's scripting language, doesn't require external native plug-ins and can be used on any platform Unity supports.**

Installation

To install Fabric you simple need to import the Fabric.package into your project.

The package comes with two DLL assemblies, documentation and a set of tutorials.

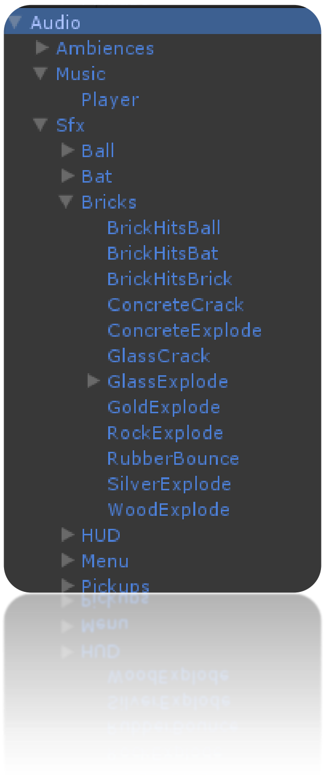
Getting started

Here we will outline some the key concepts and components of Fabric.

Fabric audio hierarchy

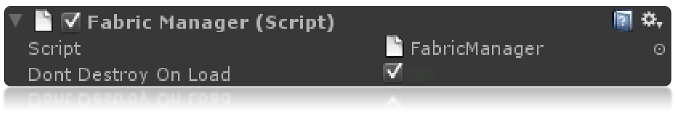
Fabric uses Unity's object based hierarchical structure in order to build complex audio behaviours.

Each game object must have a Fabric component within it in order to be able to expand the hierarchy.



FabricManager

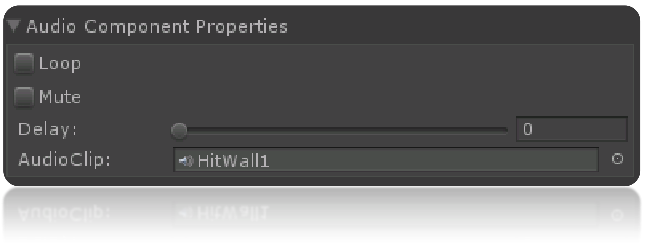
Fabric manager component must always be at the top of the hierarchy. Its main responsibility is to manage the components and their life span.


Fabric Manager (Script)
Script FabricManager
Dont Destroy On Load ✓

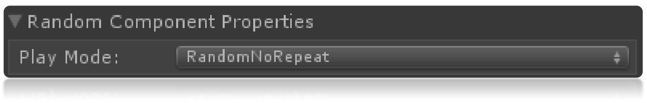
Components

Components provide the building blocks that allow you to create sophisticated audio designs using any combination of the following available components:

Audio Component: The simplest component that extends Unity's existing audio source component functionality. This component is always the last node in the hierarchy.



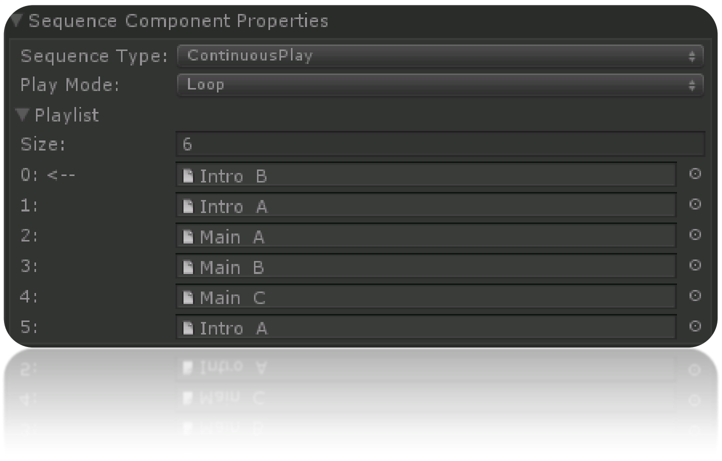
Audio Component Properties
Loop
Mute
Delay: 0
AudioClip: HitWall1



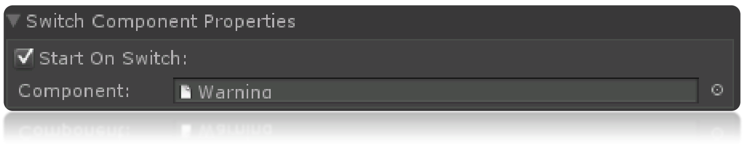
Random Component Properties
Play Mode: RandomNoRepeat

Random Component: Triggers a series of components in a random order or in a way that avoids repetition.

Sequence Component: Triggers a series of components defined by a playlist.

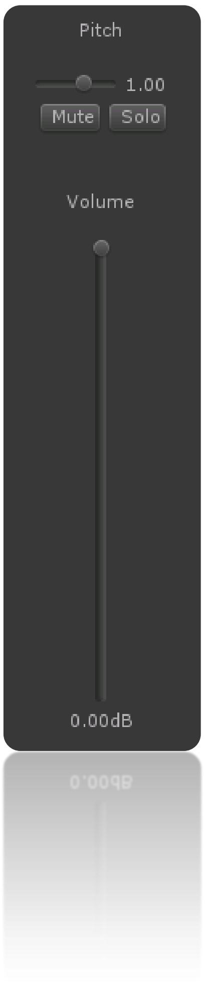


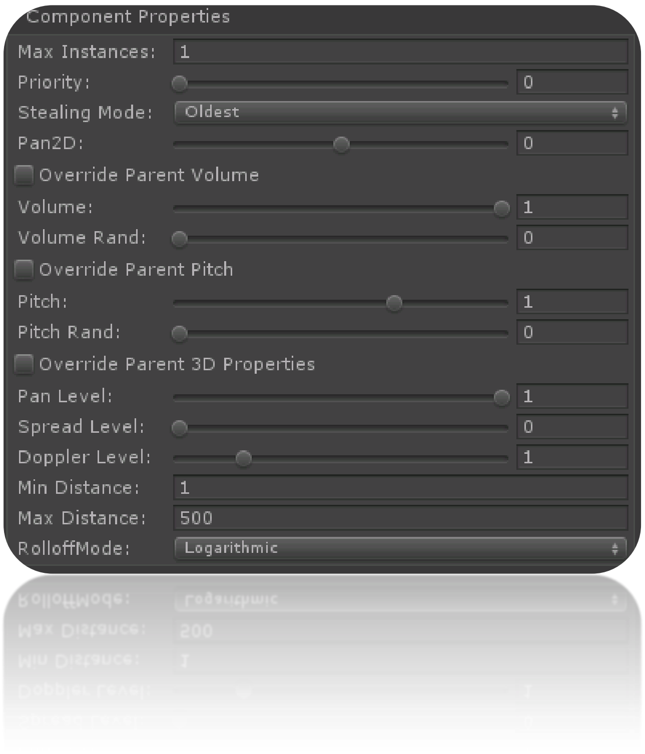
Sequence Component Properties
Sequence Type: ContinuousPlay
Play Mode: Loop
Playlist
Size: 6
0: <-- Intro B
1: Intro A
2: Main A
3: Main B
4: Main C
5: Intro A



Switch Component: Determines which component to trigger from an option that is set by the game. The option is **ALWAYS** the name of the component that is required to trigger.

Group Component: Controls the volume and pitch properties of all the sounds in its hierarchy.





Component Properties

Max Instances:	1
Priority:	0
Stealing Mode:	Oldest
Pan2D:	0
Override Parent Volume	
Volume:	1
Volume Rand:	0
Override Parent Pitch	
Pitch:	1
Pitch Rand:	0
Override Parent 3D Properties	
Pan Level:	1
Spread Level:	0
Doppler Level:	1
Min Distance:	1
Max Distance:	500
RolloffMode:	Logarithmic

All components inherit the same core properties. All of these properties are propagated through their hierarchy and are either added or multiplied with their parent properties. However, it is possible to override some of the parent properties and use their own values instead.

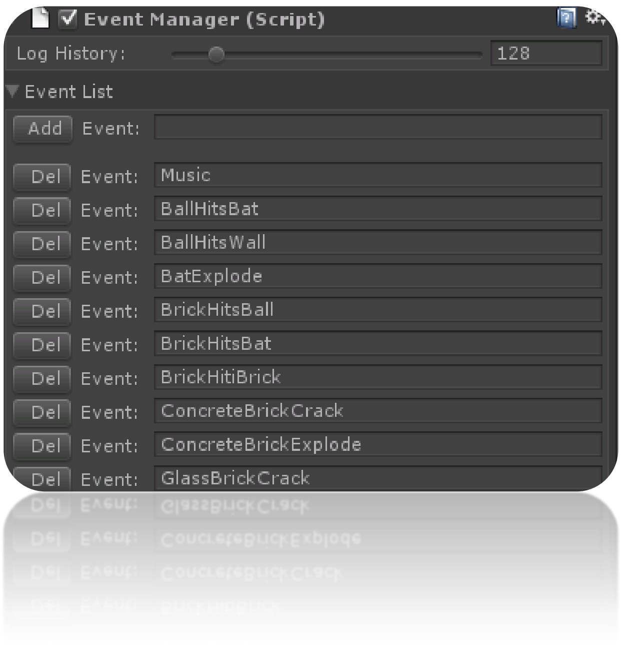
Events

Fabric's event based system allows events to be used by the game in order to drive the audio. Each event can be set to perform an action chosen from a list such as: play, stop, set volume, set pitch, set parameters. Events can be implemented very easily in a game either in the editor or by code even before any audio authoring is done. This method allows creating and re-iterating on the audio content of the game quickly and efficiently.

EventManager

The event manager component is responsible for managing the list of events as well as the flow of events between the game and the fabric components.

NOTE: EventManager component needs to be located in the same game object as the Fabric Manager component. Fabric will automatically add it at runtime if one is not available but it will not be possible to store the event list.



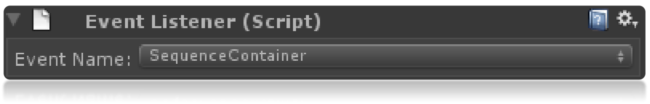
Event Manager (Script)

Log History: 128

Event List

Add	Event:	
Del	Event:	Music
Del	Event:	BallHitsBat
Del	Event:	BallHitsWall
Del	Event:	BatExplode
Del	Event:	BrickHitsBall
Del	Event:	BrickHitsBat
Del	Event:	BrickHitiBrick
Del	Event:	ConcreteBrickCrack
Del	Event:	ConcreteBrickExplode
Del	Event:	GlassBrickCrack

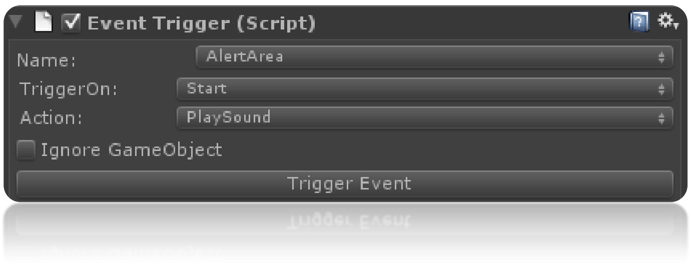
Listener


Event Listener (Script)
Event Name: SequenceContainer

When a listener component is inserted in the same game object as a component component it allows for the component to be triggered by the game. The listener simply waits and listens for a specific event name and only when it receives that name it will respond and perform an action. Furthermore, a listener can accept and set parameters if they are supported by the component.

Trigger

The Event trigger component is responsible for sending an event into Fabric with a specific action (i.e. Play, Stop, Set Volume/Pitch) and parameter values. Every component that is listening for that event within the Fabric hierarchy will respond accordingly.

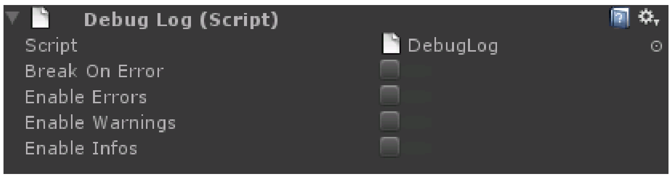

Event Trigger (Script)
Name: AlertArea
TriggerOn: Start
Action: PlaySound
Ignore GameObject
Trigger Event

An even trigger can be set to trigger when certain game object function is called or by code through the event manager.

Debug log

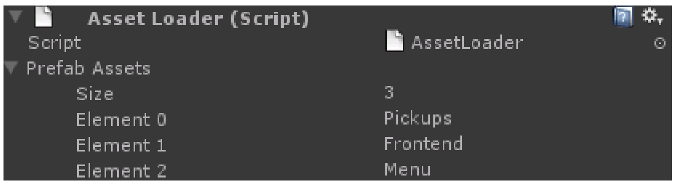
DebugLog provides the option to choose which type of messages will be displayed in the console output therefore reduces the amount of information displayed.

NOTE: The DebugLog component needs to be located in the same game object as the Fabric Manager component. Fabric will automatically add it at runtime if one is not available but it will not be possible to store it’s properties.


Debug Log (Script)
Script DebugLog
Break On Error
Enable Errors
Enable Warnings
Enable Infos

AssetLoader

Asset loader is a component that loads and unloads prefabs assets that contain Fabric components. When the component is started it loads the prefab assets from a list and when it is destroyed it unloads them.



Asset Loader (Script)

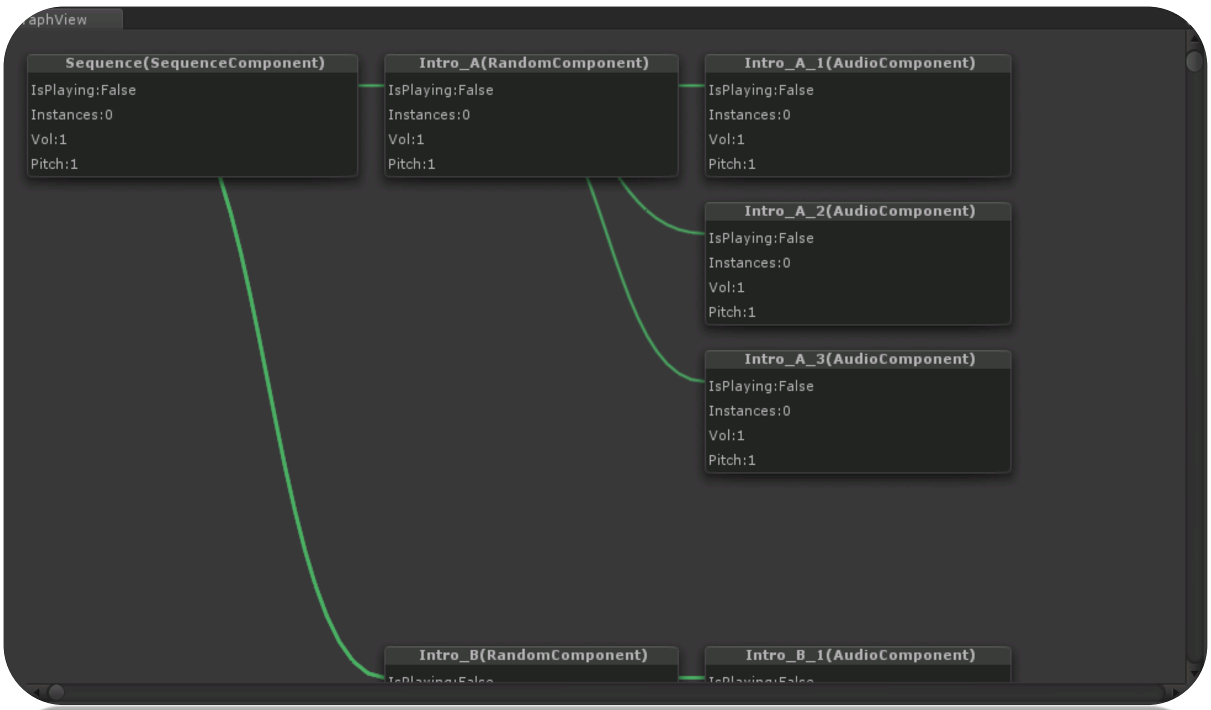
Script	AssetLoader
Prefab Assets	
Size	3
Element 0	Pickups
Element 1	Frontend
Element 2	Menu

Interfaces

Fabric provides a number of custom user interfaces that allows quickly iterating and focusing on the audio functionality that is important.

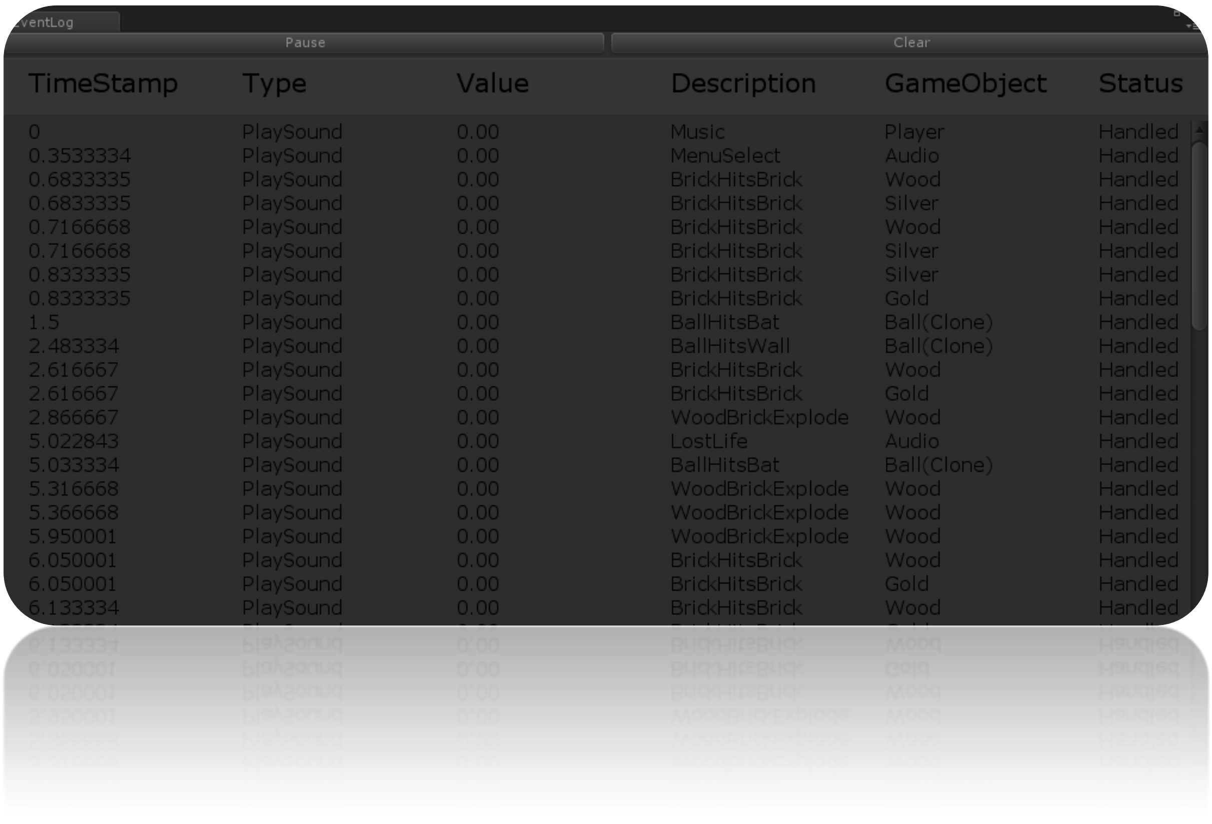
Graph View

Provides a flat view of the whole component hierarchy with runtime information of their state.



Event Log

Event log view allows monitoring the flow of events and identifying potential problems or missing events. It is possible at any moment to pause or clear the event log.

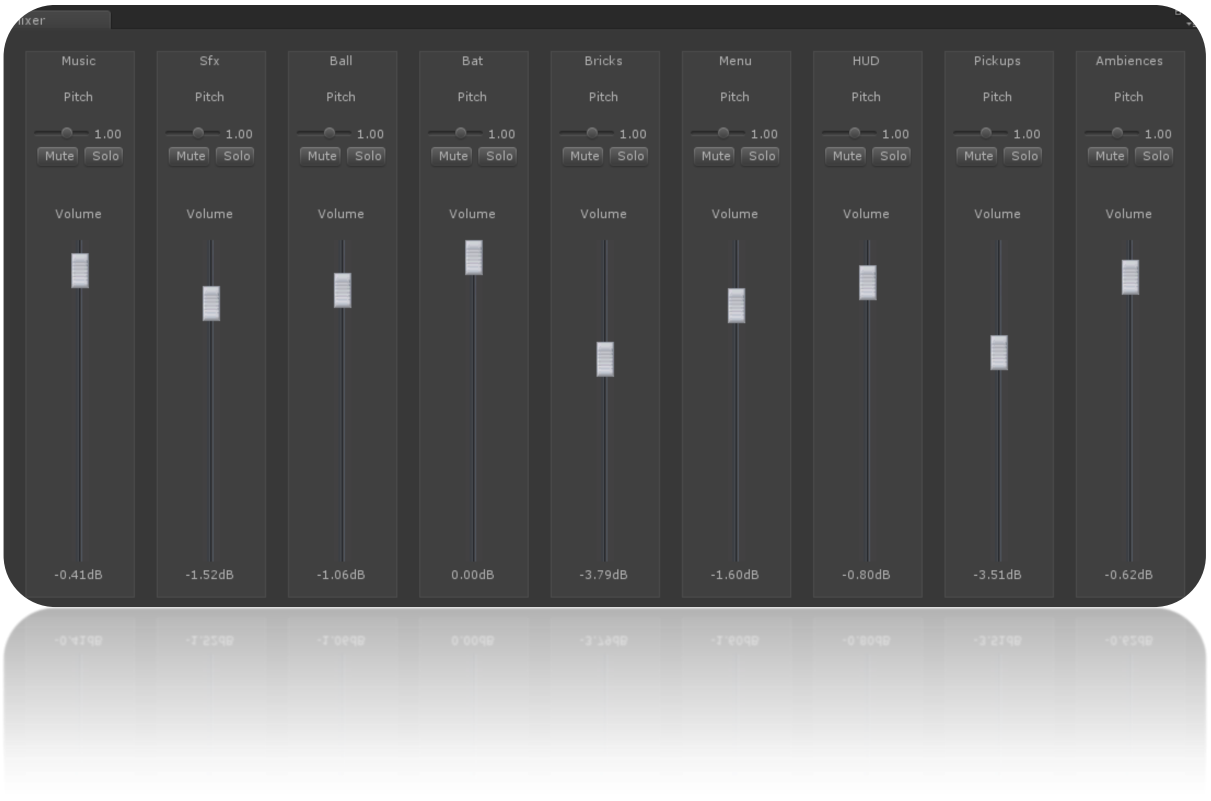


TimeStamp	Type	Value	Description	GameObject	Status
0	PlaySound	0.00	Music	Player	Handled
0.3533334	PlaySound	0.00	MenuSelect	Audio	Handled
0.6833335	PlaySound	0.00	BrickHitsBrick	Wood	Handled
0.6833335	PlaySound	0.00	BrickHitsBrick	Silver	Handled
0.7166668	PlaySound	0.00	BrickHitsBrick	Wood	Handled
0.7166668	PlaySound	0.00	BrickHitsBrick	Silver	Handled
0.8333335	PlaySound	0.00	BrickHitsBrick	Silver	Handled
0.8333335	PlaySound	0.00	BrickHitsBrick	Gold	Handled
1.5	PlaySound	0.00	BallHitsBat	Ball(Clone)	Handled
2.483334	PlaySound	0.00	BallHitsWall	Ball(Clone)	Handled
2.616667	PlaySound	0.00	BrickHitsBrick	Wood	Handled
2.616667	PlaySound	0.00	BrickHitsBrick	Gold	Handled
2.866667	PlaySound	0.00	WoodBrickExplode	Wood	Handled
5.022843	PlaySound	0.00	LostLife	Audio	Handled
5.033334	PlaySound	0.00	BallHitsBat	Ball(Clone)	Handled
5.316668	PlaySound	0.00	WoodBrickExplode	Wood	Handled
5.366668	PlaySound	0.00	WoodBrickExplode	Wood	Handled
5.950001	PlaySound	0.00	WoodBrickExplode	Wood	Handled
6.050001	PlaySound	0.00	BrickHitsBrick	Wood	Handled
6.050001	PlaySound	0.00	BrickHitsBrick	Gold	Handled
6.133334	PlaySound	0.00	BrickHitsBrick	Wood	Handled

Mixer

The mixer window displays the group components in the hierarchy. From this view it is possible to change the volume or pitch properties of individual components.

The mixers Mute and Solo functionality allows to easily to mix and balance the audio levels in a game.



Music	Sfx	Ball	Bat	Bricks	Menu	HUD	Pickups	Ambiences
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
-0.41dB	-1.52dB	-1.06dB	0.00dB	-3.79dB	-1.60dB	-0.80dB	-3.51dB	-0.62dB

DSP Effects

Fabric does not manage DSP effects at the moment, though plans are made to support this in the future. However, this does not mean that DSP effect components cannot be used with Fabric components.

In order to use the effects all that needs to be done is to add an AudioSource and the desire effect components on the game object that has the AudioComponent component.

Quick walkthrough

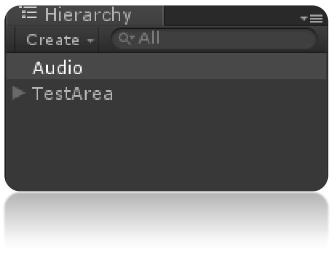
This is a quick walkthrough explaining how to create and trigger a simple audio component in Fabric.

Setting up the scene

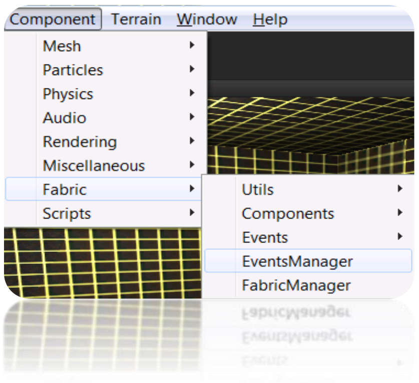
First we are going to start with a new empty Unity project in which we are going to place our audio assets, scene and Fabric hierarchy.

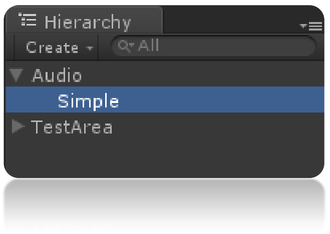
Creating the audio hierarchy

- Create a game object which is going to be the root of Fabric and give it a meaningful name (i.e. Audio, GameAudio etc.)

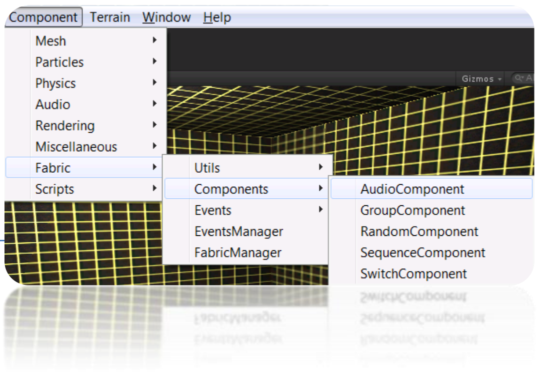


- Highlight the game object and from the Components->Fabric menu add FabricManager and EventManager components.



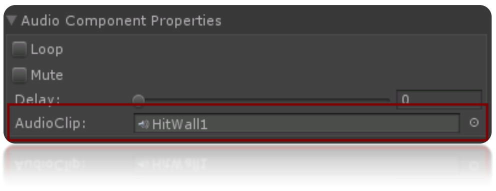


- Create another game object below the root and call it Simple.



- From the Components->Fabric->Components menu selection add the AudioComponent component.

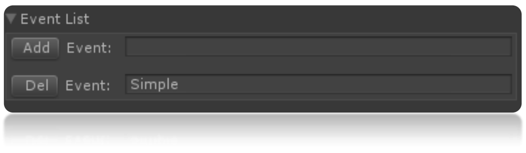
- Next step is to assign the audio clip property of the audio, this could be done either by dragging and dropping the audio file in the AudioClip property or by clicking on it.

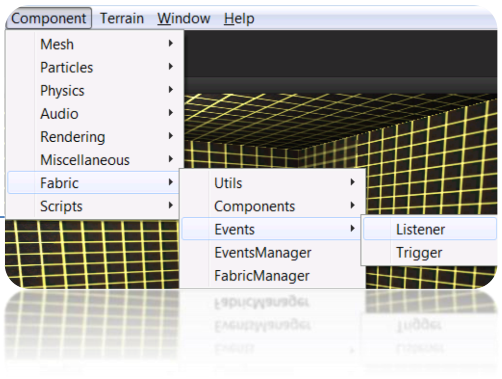


Creating the events

Each component needs to have a listener with a specific event name that it will listen to and respond accordingly.

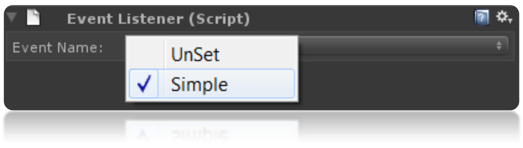
- In the EventManager enter the name of the event and click on the Add button to add it into the list of available events.





- Highlight the Simple game object and from the Components->Fabric->Events add the EventListener component.

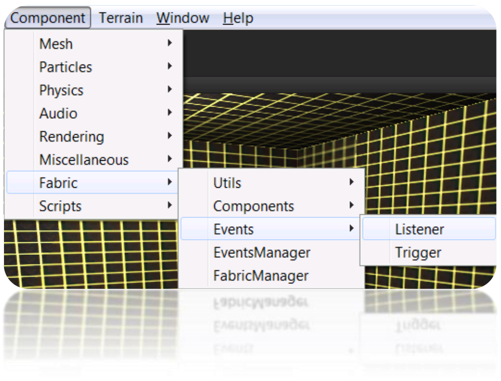
- Next step is to select the event name from the drop down menu.



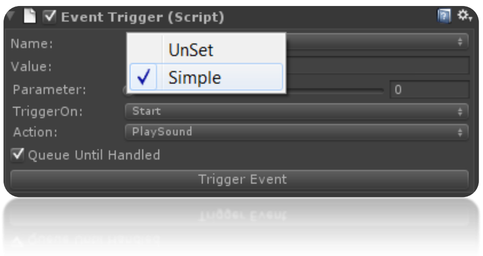
Event Listener (Script)

Event Name: Simple

- Now create another empty game object, outside of the audio hierarchy, and from the Components->Fabric->Events menu selection add an EventTrigger.



- In the event name drop down menu select the same event name as the listener.



Event Trigger (Script)

Name: Simple

Value:

Parameter: 0

TriggerOn: Start

Action: PlaySound

✓ Queue Until Handled

- And that’s it, when you run the game it will automatically trigger the audio component and you should be able to hear a sound. It is also possible to trigger the audio again by clicking on the trigger button.