

Koreographer 快速入门指南

对于 v1.6.1



目录

概述	3
最低要求元素	3
指令概要	3
详细演练	3
打开和查看 Koreography 编辑器	3
编导	4
创建和配置 Koreography 创建和配置	4
KoreographyTrack 创建和配置	5
KoreographyEvents	5
事件类型	6
有效负载类型	6
在 Unity 场景中设置 Koreographer 创建处理	8
KoreographyEvents 的组件	10
事件 ID	12
回调函数	12
完成的组件脚本	13
运行和测试 Koreographer 功能	14

概述

为了演示在场景中使用 Koreographer，我们将使用尽可能少的步骤在准系统 Unity 场景中创建基本的 Koreographer 设置。

最低要求元素

- 当 *编剧零件*。
- 一个实现 IKoreographedPlayer 接口（示例包括提供的 *简单音乐播放器* 和 *多音乐播放器成分*）。
- 至少一个 *音频剪辑* (与 [负载类型](#) 设置为 *加载时解压*) .这可以是任何音频剪辑。
- 至少一个 *韩剧资产* 又包含...
- 至少一个 *韩国摄影追踪资产* 反过来具有...
- 一个或多个 *韩国摄影活动*。
- 一个或多个 *单一行为* 已注册的组件 [事件编号](#) 的 *韩国摄影追踪*。

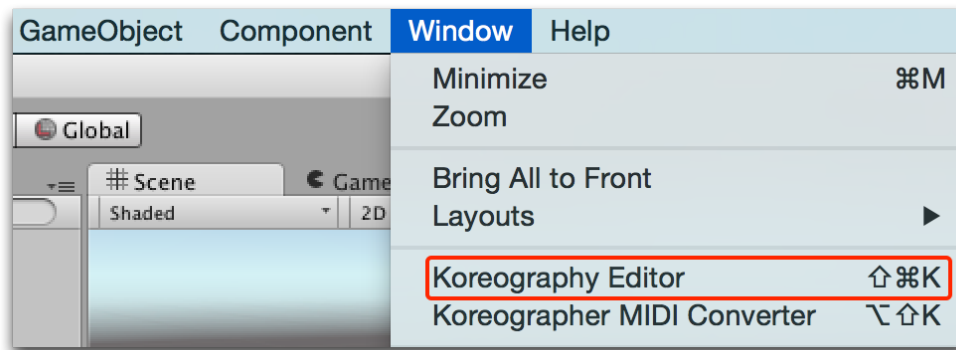
指令概要

1. 打开一个现有的 Unity 项目或创建一个新项目。
2. 导入 Koreographer Unity 包（一般可以通过 [资产商店](#)）。
3. 打开 Koreography 编辑器。
4. 创建一个 *韩剧资产* 文件。
5. 关联 *韩剧* 与一个 *音频剪辑*。
6. 新建 *韩国摄影追踪*。
7. 定义事件 ID *韩国摄影追踪*。
8. 创建一个或多个 *韩国摄影活动*。
9. 创建一个 *你好韩国摄影师* 单行为。
10. 使用以下组件设置和测试 Unity 场景：
 - 一种。 *编导*
 - 湾。 *简单音乐播放器*
 - C。 *你好韩国摄影师*

详细演练

打开和查看 Koreography 编辑器

1. 打开 Unity 并在新项目或现有项目中创建新场景。
2. 导入 Koreographer Unity 包（一般可以通过 [资产商店](#)）。
3. 导入 Koreographer Unity 包后，**编导** 菜单条目被添加到 **窗户** 菜单。选择该菜单项以打开 Koreography 编辑器。



Koreography 编辑器可以从窗口菜单打开

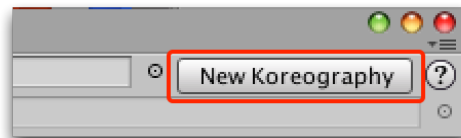
编导

Koreography Editor 是用于创建和编辑的主要工具：

- **韩剧：** 关联一个资产 音频剪辑与一群 韩国摄影追踪s 和速度图（通过速度部分定义）。
- **KoreographyTracks：** 一个序列 韩国摄影活动 沿音频时间线定义。
- **韩国摄影活动：** 一个事件，无论是瞬时的 (OneOff) 还是覆盖时间跨度 (Span)，都可以选择包含 [有效载荷](#)

创建和配置 Koreography

4. 现在Koreography Editor已经打开了，我们必须创建我们的第一个 韩剧资产。编辑器窗口的右上角是按钮，允许您加载现有的 韩剧资产或创建一个新的。

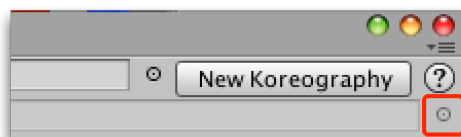


创建并加载一个新的 韩剧资产与新朝鲜文学 按钮

点击**新朝鲜文学** 按钮，为新的资产文件命名**MyFirstKoreography**，并将其保存在您的 Unity 项目中。

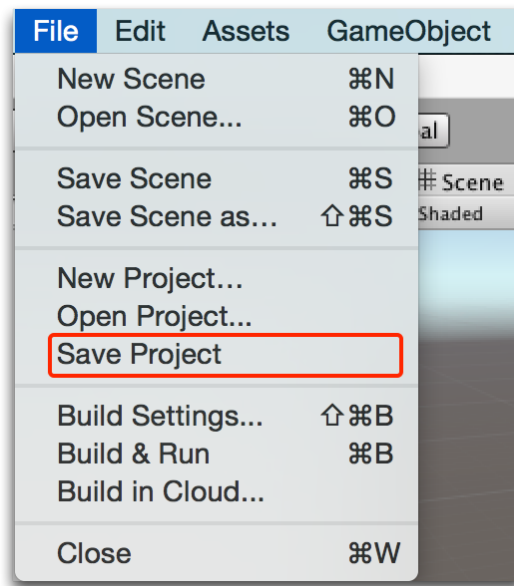
5.现在我们有了一个 韩剧加载资产，我们必须关联一个一种
Unity 加载小部件 “⊙” 在右侧 音频剪辑字段并选择您的音频。

音频剪辑用它。选择



使用 Unity 加载小部件 “⊙” 加载一个 音频剪辑

加载后，通过从“文件”菜单中选择“保存项目”来保存资产。



在 Unity 中保存对资产文件的更改 **保存项目** 菜单项

创建和配置 KoreographyTrack

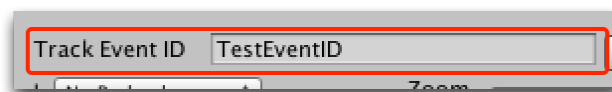
6.现在我们有了一个 **韩剧** 具有关联的资产 **音频剪辑**，我们必须创建一个 **韩国摄影追踪**，它将包含我们的事件。
为此，请选择**新的** Koreography Editor 窗口右侧的 Track 按钮。



创建并加载一个新的 **韩国摄影追踪** 资产与**新的** 按钮

为新的资产文件命名**MyFirstKoreographyTrack** 并将其保存在您的 Unity 项目中。

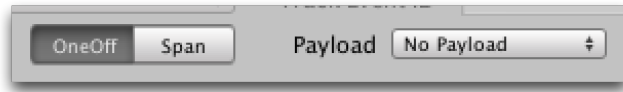
7.当我们创建 **韩国摄影追踪**，事件 ID 设置为资产的名称（非唯一字符串 ID）。事件 ID 是侦听器脚本用于注册来自的事件的键 **编剧** 在运行时。把ID从文件名改成**测试事件ID**。



设置跟踪事件 ID 的 **韩国摄影追踪** 到 “TestEventID”

创建和配置 KoreographyEvents

现在我们有了创建的所有先决条件 **韩国摄影活动**。在我们这样做之前，我们将配置**结构体** 我们希望创建的事件。要配置事件类型和有效负载类型，请查看波形正上方的屏幕右侧。

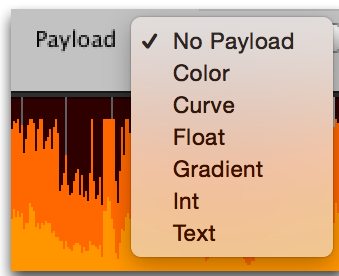


新创建的事件类型和有效负载类型设置 *韩国摄影活动*

事件类型

- **一次性：** 如果**一次性** 按钮在期间被按下 *韩国摄影活动*创作，[一次性](#) 事件被创建。
- **跨度：** 如果**跨度** 按钮被按下，事件将[跨度](#) 沿着音频时间线的时间范围。事件在按下事件创建键时开始，在按下的键被释放时结束。

有效负载类型



内置负载类型

- **无负载：** 此事件本质上是一条非描述性消息；一个触发器。
- **Pro 颜色：** 本次活动承载了 *颜色* 事件订阅者可以检索其中的值。
- **曲线：** 与 *颜色* 除了它带有一个 *动画曲线* 目的。
- **漂浮：** 与 *颜色* 除了它带有一个数字。
- **Pro 坡度：** 与 *颜色* 除了它带有一个 *渐变* 目的。
- **诠释：** 与 *颜色* 除了它带有一个整数。
- **文本：** 与 *颜色* 除了它带有文本。
- **Pro 光谱：** 与 *颜色* 除了它携带一组频谱。此 Payload 不会出现在 Payload 列表中，只能通过 **分析设置** 窗户。

韩国摄影活动 可以用 **绘图工具** 或播放期间的键盘。对于本指南，我们将在播放期间使用键盘添加它们。

8. 要使用键盘创建事件，我们必须播放 *音频剪辑* 使用 Koreography Editor 窗口左侧的音乐播放器按钮。



音乐播放器按钮

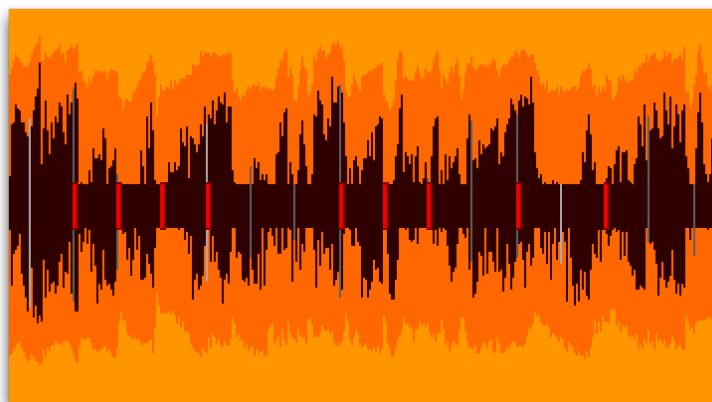
播放音频后，我们按 **乙, 返回**，或 **进入** 连同中的点 *音频剪辑* 我们想在哪里生成一个事件。 *韩国摄影活动* 是使用生成的 [活动设置](#) 只是

在波形视图上方，将显示为垂直的红色条一次性事件部分跨度。

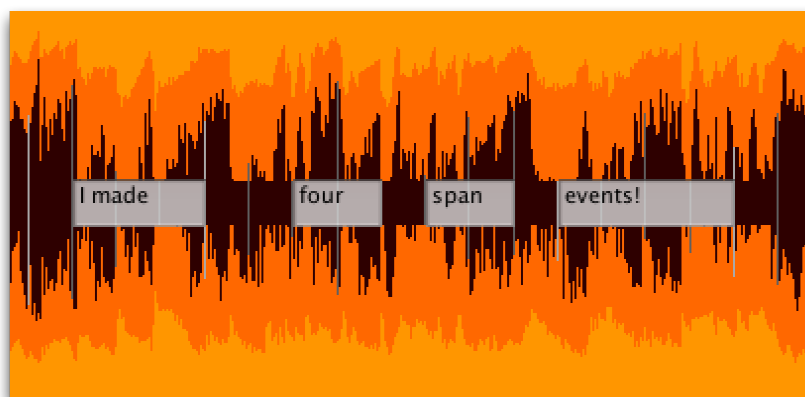
通风口或矩形

笔记： 波形视图左下角的正下方是 *捕捉到节拍* 环境。这启用音乐量化，默认开启。取消选中该框以允许自由格式的音乐标记。

与 *捕捉到节拍* 选中后，Koreography Editor 将使用 *速度部分设置* 以确定节拍在音乐中的位置。节拍位置被描绘为波形后面的垂直白线。用 *调整节奏速度部分设置* 将节拍与音乐对齐的工具。

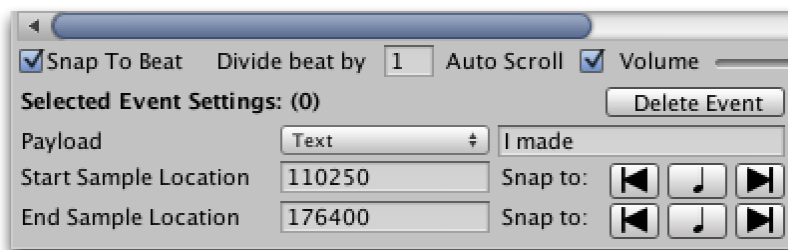


一次性事件



跨度事件

在 Koreography Editor 的左下角创建后，可以手动更改或调整选定事件的有效负载。您可以随时查看活动的 *有效载荷*，或缺乏，通过选择事件并参考 *选定的事件设置* 在左下角。



示例文本有效负载设置

有关事件类型和有效负载配置的更多详细信息，请参阅**Koreographer 用户指南** 本文件随附的。

在 Unity 场景中设置 Koreographer

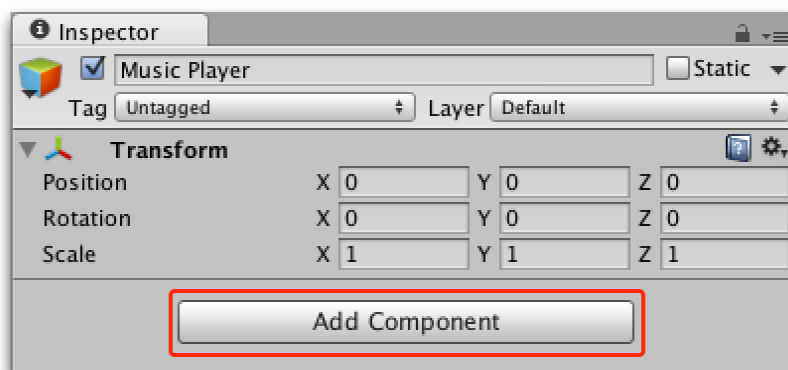
9. 现在我们已经创建了一个**韩剧资产**和相关联的**音频剪辑**，一个**韩国摄影追踪**和**韩国摄影活动** 有了它，我们可以继续设置我们的场景。我们的第一步是创建一个**游戏对象**附上我们的**简单音乐播放器**和**编剧成分**。

一种。创建一个空的**游戏对象**为这个场景重命名**音乐播放器**。



选择**创建空** 从**游戏对象**菜单

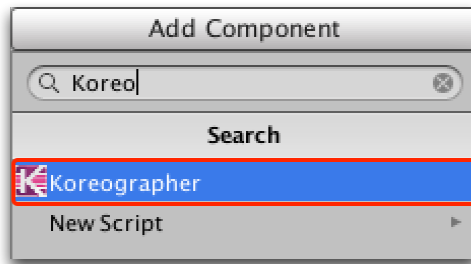
湾。选择**游戏对象**在层次结构中并选择一种**dd 组件** 在检查员中。



当添加组件 按钮

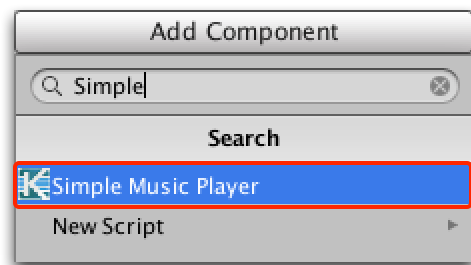
C. 类型**编导** 进入搜索字段并选择**K**
游戏对象.

地理学家s 脚本将其添加到此



添加 **编剧**零件

d. 我们现在将重复最后两个步骤来添加**简单音乐播放器**此对象的组件。添加**简单音乐播放器**脚本也会添加所需的**音频源**零件。



添加 **简单的音乐播放器**零件

10.现在我们有了一个**游戏对象**在具有跟踪和触发事件的两个必要类的场景设置中，我们需要分配**韩剧**我们配置的。**分配韩剧**资产将包括**韩国摄影轨道**s ,这**K**

地质学事件s ,和**音频剪辑**我们分配给它。

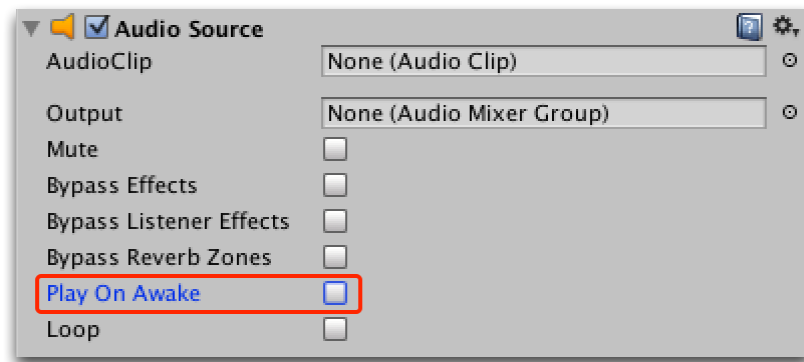
一种。我们需要分配**MyFirstKoreography** 资产 **简单音乐播放器**零件。这可以通过拖动我们的**MyFirstKoreography** 将资产放入空的表单字段或通过选择 Unity 加载小部件 “○” 在空白表单字段的右侧，在结果弹出窗口中选择资产选项卡，然后选择正确的资产。



设置 **Koreography** 字段**简单的音乐播放器**

笔记: 无需设置**音频剪辑**上**音频源**零件。当**简单音乐播放器**
在游戏中为我们处理!

湾。我们建立我们的最后一步 编导场景中的对象是取消选中 音频源 醒着玩 财产。



取消勾选醒着玩

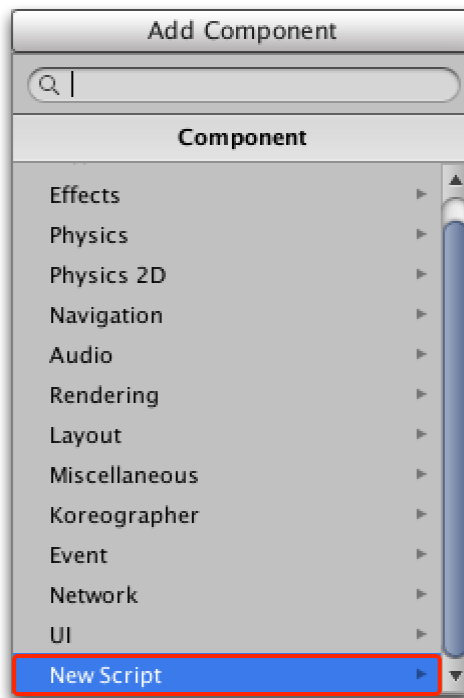
创建一个处理 KoreographyEvents 的组件

我们现在有一个成熟的 编导对象 准备好跟踪和触发事件。我们只需要一个 游戏对象 能够订阅这些事件的场景。要添加此功能，我们需要创建一个新组件。

11.再创建一个空 游戏对象 场景并命名 测试事件订阅者。

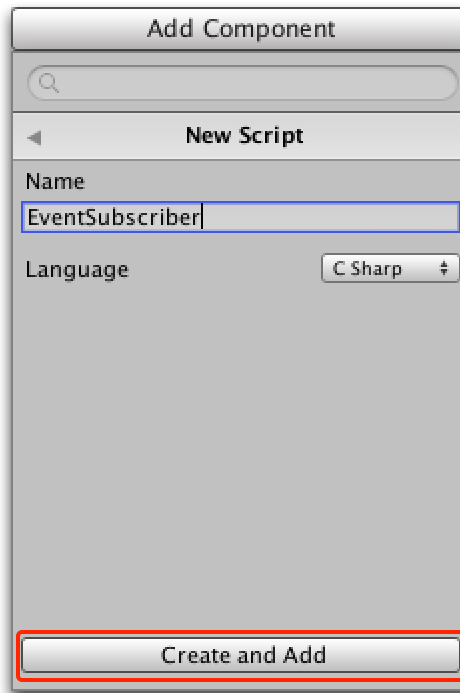
12.在新建的C#脚本组件中添加新的 游戏对象. 去做这个：

一种。选择 添加组件 下 测试事件订阅者 Unity Inspector 中的属性并选择 新脚本 菜单项。



选择新脚本

湾。为新脚本命名 事件订阅者 和 将语言切换为 C#。确认这些设置并创建/添加脚本到 测试事件订阅者 通过选择对象 创建和添加。



创建并添加活动订阅者脚本

你会看到事件订阅者脚本现在附加到游戏对象以及在 Unity 的项目选项卡中。

13. 我们现在需要编辑脚本，以便它可以响应来自我们的事件朝鲜文学。在任一位置双击它，脚本将在您首选的脚本编辑器（这可能是 MonoDevelop 或 Visual Studio）中打开。

一种。起初，你的脚本会像这样是空白的。

```
使用统一引擎；
使用System.Collections；

公开 班级活动订阅者：单一行为{

    //使用这个进行初始化无效开始 () {

    }

    // 每帧调用一次更新无效更新 () {

    }

}
```

湾。订阅编剧对于事件很简单：我们访问单例实例（全局对象的花哨名称）编剧并告诉它我们想要订阅什么事件。

C。在订阅事件时，我们必须考虑两件事：

- 的事件编号 和

■ 的[回调函数](#) 触发事件时触发。

事件 ID

此事件 ID 是创建时设置的内容 *韩国摄影追踪* 本身作为 *韩国摄影追踪* 文件（首次创建时默认）或创建后自定义名称。出于我们的目的，我们将其设置为 **测试事件ID**。事件 ID 允许我们创建可以同时触发的许多不同的 Koreography 轨道。

回调函数

回调函数依赖于函数委托。函数委托定义了回调的结构：它的参数和它们的类型——它的签名！我们将创建一个与 Koreographer 系统的委托使用相同结构的函数。当我们注册这个函数时 *编剧*，我们还将注册事件 ID。

编剧 定义了两个函数委托，我们可以在创建回调函数时使用：

```
公开 代表 无效 KoreographyEventCallback(韩国摄影活动韩国事件)； 公开 代表 无效
KoreographyEventCallbackWithTime(韩国摄影活动韩国活动, 整数
    采样时间, 诠释样本三角洲, 三角切片增量切片)；
```

第一个委托函数 `KoreographyEventCallback` 定义了一个返回类型为 `void` 的函数和一个 *韩国摄影活动* 范围。这用于当您只希望接收事件对象本身时。

第二个委托函数 `KoreographyEventCallbackWithTime` 定义了一个函数，返回类型为 `void`，三个参数为 *韩国摄影活动*，二 *诠释* 小号（*采样时间* 和 *样本三角洲*），和一个 *增量切片* 目的。这用于当您希望接收事件以及有关事件发生在轨道中的位置的一些信息时。当 *采样时间* 参数是样本中音频的当前时间 *韩国摄影活动* 发送时 *样品三角洲* 将是自上一帧以来的样本总数（类似于如何 *时间.deltaTime* 提供自上一帧以来的规范时间）。欲了解更多信息 *三角切片* 对象，见 *Koreographer 用户指南*。

14. 为了使用 Koreographer API，我们需要将命名空间添加到脚本中。添加 “[使用](#) `SonicBloom.Koreo`；” “下面” [使用](#) 统一引擎；“指令：

```
使用统一引擎；
使用SonicBloom.Koreo；
```

15. 创建匹配第一个委托的单个函数（只有单个参数）*韩国摄影活动*。在函数的主体中，我们将添加一个 `Debug` 语句来打印有关接收事件的消息。如下所示，函数的结构 *FireEventDebugLog* 匹配的 `KoreographyEventCallback` 功能委托。

```

无效FireEventDebugLog(韩国摄影活动韩国事件) {

    调试.Log(“Koreography 事件被解雇了。”);
}

```

16. 我们测试前的最后一个任务是注册Event ID和回调函数 *FireEventDebugLog*. 注册事件是通过调用 *编剧* 单身的 *注册活动* 功能。每个对象实例都会这样做一次，因此放置它的好地方是 *开始* 功能。

```

// 使用这个进行初始化 无效开始 ()

{
    编剧.Instance.RegisterForEvents(“测试事件ID”, 火事件调试日志);
}

```

完成的组件脚本

完成版 *活动订阅者* 脚本看起来像这样。保存文件并关闭脚本编辑器。

```

使用统一引擎;
使用SonicBloom.Koreo;

公开 班级活动订阅者: 单一行为 {

    // 使用这个进行初始化 无效开始 ()

    {
        Koreographer.Instance.RegisterForEvents(“测试事件ID”, 火事件调试日志);
    }

    无效FireEventDebugLog(KoreographyEvent koreoEvent) {

        调试.Log(“Koreography 事件被解雇了。”);
    }
}

```

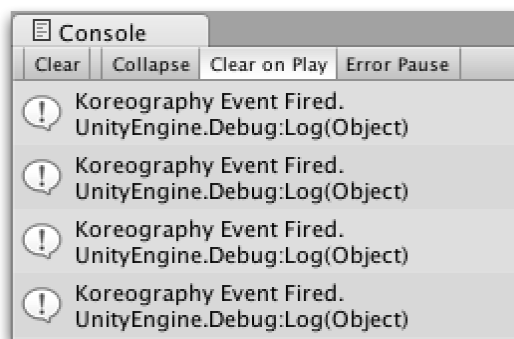
运行和测试 Koreographer 功能

17. 现在我们回到了 Unity，我们可以测试以确保我们的事件被接收。按播放按钮。



团结的播放场景 按钮

随着音频播放和事件被触发，您应该会在控制台中看到日志语句 *测试事件订阅者*。



观察控制台的调试输出！