



# Cube 集成

Wwise

Copyright © 2025 Audiokinetic Inc. All rights reserved.

This document (whether in written, graphic or video form) is supplied as a guide for the Wwise® product. This documentation is the property of Audiokinetic Inc. (“Audiokinetic”), and protected by Canadian copyright law and in other jurisdictions by virtue of international copyright treaties.

This documentation may be duplicated, reproduced, stored or transmitted, exclusively for your internal, non-commercial purposes, but you may not alter the content of any portion of the documentation. Any copy of the documentation shall retain all copyright and other proprietary notices contained therein.

The content of the Cube Integration documentation is furnished for information purposes only, and its content is subject to change without notice. Reasonable care has been taken in preparing the information contained in Cube Integration, however, we disclaim all representations, warranties and conditions, whether express, implied or arising out of usage of trade or course of dealing, concerning this documentation and assume no responsibility or liability for any losses or damages of any kind arising out of the use of this guide or of any error or inaccuracy it may contain, even if we have been advised of the possibility of such loss or damage.

Wwise®, Audiokinetic®, Actor-Mixer®, SoundFrame® and SoundSeed® are registered trademarks, and Master-Mixer™, SoundCaster™ and Randomizer™ are trademarks, of Audiokinetic. Other trademarks, trade names or company names referenced herein may be the property of their respective owners.

---

## Table of Contents

1. 关于 Wwise Cube 集成 .....	1
2. 安装和运行 Cube Demo .....	2
3. 了解如何将声音引擎集成到《Cube》中 .....	6
4. 了解 Wwise 工程 .....	8
5. 调整声音设计 .....	10
6. 需要帮助? .....	12

---

## Chapter 1. 关于 Wwise Cube 集成

Audiokinetic 将 Wwise 声音引擎集成到了魔改版开源第一人称游戏《Cube》中。为顺应相关要求，此版本改用了《Wwise Adventure Game》示例游戏中的 Wwise Wizard。

Cube 集成可帮助用户更好地了解如何将 Wwise 声音引擎集成到游戏中，并提供一个方便声音设计师练习使用 Wwise 的功能游戏。之所以选择这个游戏作为示例，是因为其结构简单且在必要时比较容易修改。

本文档分为以下几个主要章节：

- [Chapter 2, 安装和运行 Cube Demo](#) – 展示如何在 Mac® 和 Windows 上安装和运行 Cube Demo（使用或不使用 Audiokinetic Launcher）。
- [Chapter 3, 了解如何将声音引擎集成到《Cube》中](#) – 展示如何将 Wwise 声音引擎集成到《Cube》中。
- [Chapter 4, 了解 Wwise 工程](#) – 阐释在 Wwise 工程中所做的设计决策。
- [Chapter 5, 调整声音设计](#) – 展示如何实时微调声音属性并利用新生成的 SoundBank 来整合这些更改。

## Chapter 2. 安装和运行 Cube Demo

Cube Demo 数据包包含以下组件：

- Audiokinetic 的 Debug、Profile 和 Release 版《Cube》游戏。
- 用于修改声音并将其添加到游戏中的 Wwise 工程。
- 《Cube》游戏引擎源代码和文档。



### Note

内容提供商发来的素材最初都是高品质的（48 kHz/16 或 24 位）。不过，为了节省下载时间和磁盘空间，有些声音素材的大小被减小了。

有关如何安装和运行 Cube Demo 的详细信息，请参阅以下章节。

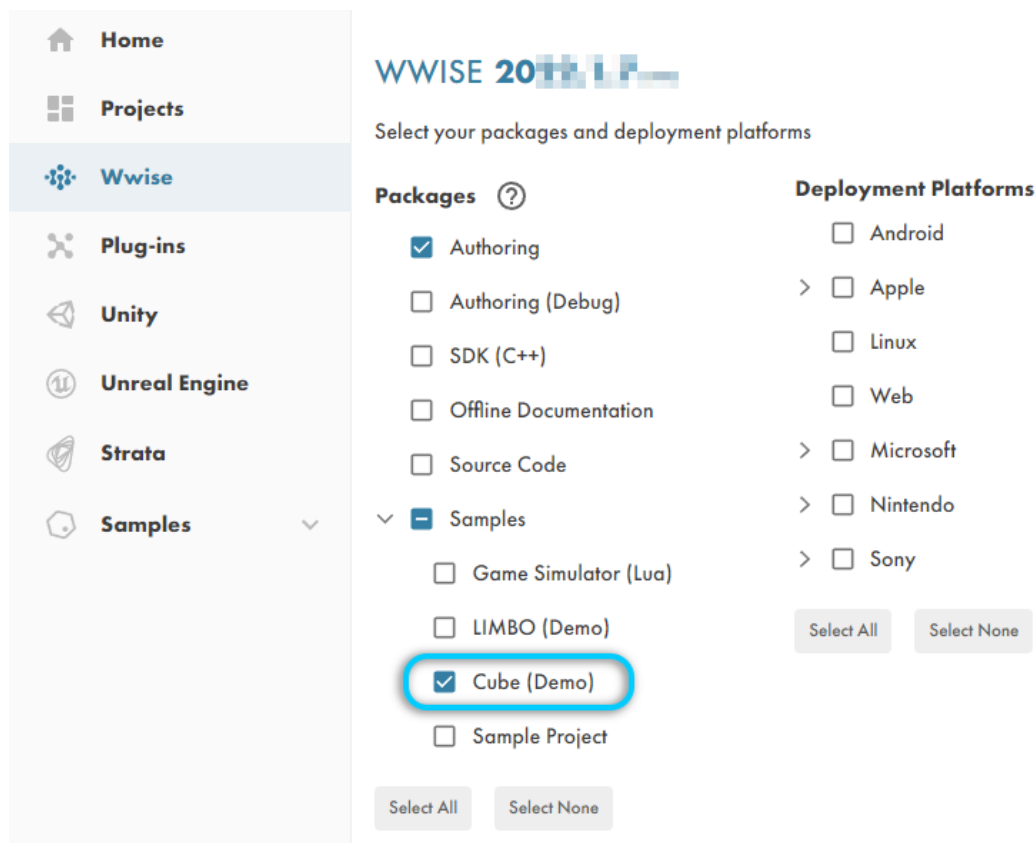
## 安装 Cube Demo

您可以通过 Audiokinetic Launcher 来安装 Cube Demo。

### 安装 Cube Demo：

- 在[安装 Wwise](#)或修改现有安装包时，展开 Samples 菜单并选中 Cube (Demo)。

若要修改现有 Wwise 安装包，请转到 Launcher 的 Wwise 页面，然后点击安装包右侧的扳手按钮并选择 Modify...。



## 通过 Launcher 运行 Cube Demo

在安装 Cube Demo 后，会将其列在 Launcher 的 Samples > Wwise Samples 页面。

### 运行 Cube Demo：


1. 在 Samples > Wwise Samples 页面，选择要安装 Cube Demo 的 Wwise 版本。
2. 点击 Run Cube 启动游戏。



## 通过 Launcher 以外的方式运行 Cube Demo

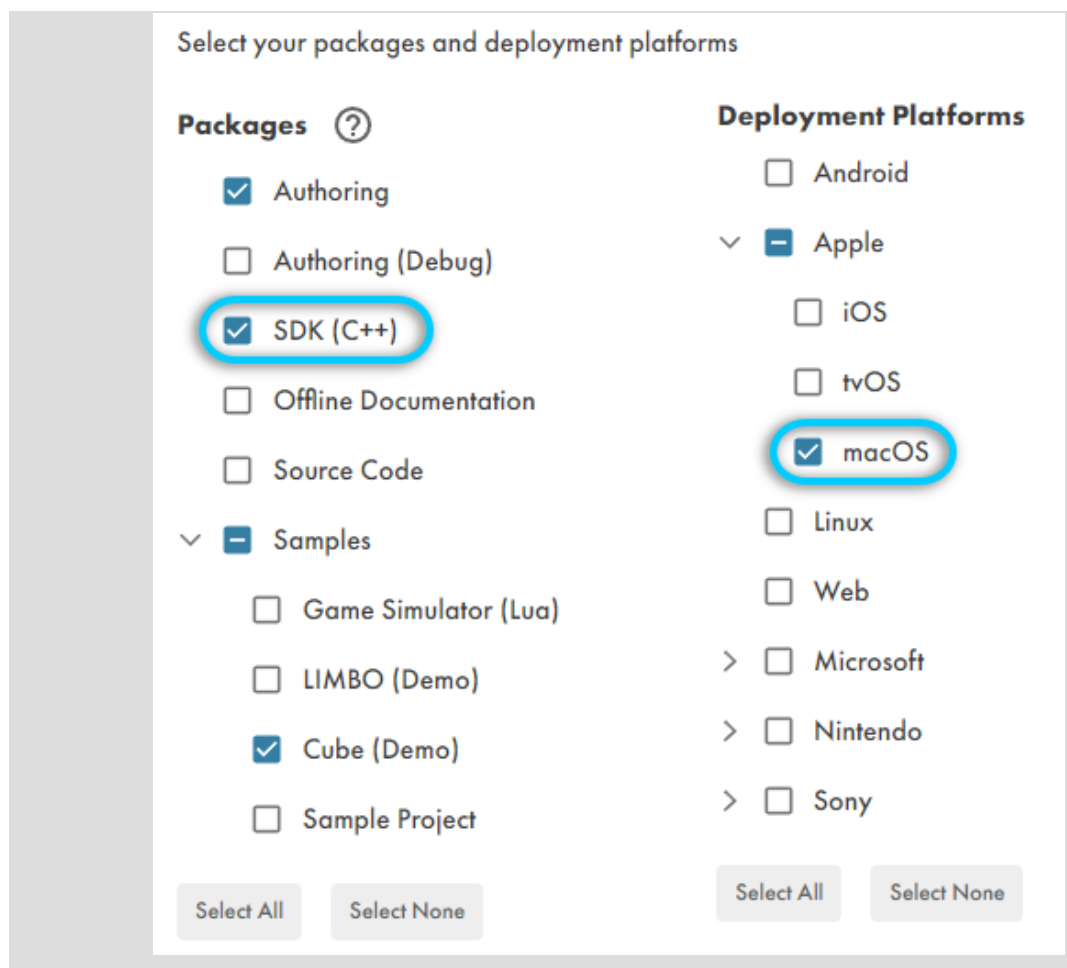
出于各种原因，您可能更倾向于通过 Launcher 以外的方式来运行 Cube Demo。对此，我们有替代的方式。不过，会因运行平台（Mac 或 Windows）而略有差异。

### 在 Mac OS X 上



**Note**

在开始之前，确保在 Wwise 安装期间选中 SDK (C++) 和 macOS 来安装 Wwise SDK for Mac。若要将这些选项添加到现有 Wwise 安装包，请转到 Launcher 的 Wwise 页面，然后点击安装包右侧的扳手按钮并选择 **Modify...**。



1. 方法 1: 使用 Shell 脚本。
  - a. 确保解压后的 Cube Demo 文件夹与安装的 Wwise SDK Mac 分发包共用同一父文件夹。比如，若 Wwise SDK 文件夹为 `${HOME}/Wwise/wwise_${version_tag}/SDK`，则 Cube Demo 根文件夹须为 `${HOME}/Wwise/wwise_${version_tag}/CubeDemo`。
  - b. 在以下位置找到可执行 Shell 脚本：`$Applications/Audiokinetic/<Wwise_version>/Cube/Mac/<mode>/bin/RunCubeDemo.sh`。其中，`<mode>` 对应于相应的模式文件夹（Debug、Profile 或 Release）。
  - c. 若 `.sh` 文件配置为通过 Terminal 打开，请在 Finder 中双击这些文件。否则，请在 Shell 脚本的目录中打开 Terminal，然后键入 `RunCubeDemo.sh` 并按下 Enter。
2. 方法 2: 在 Xcode 下运行。
  - a. 确认是否已安装 Xcode。若 Applications 文件夹中没有 Xcode，请从 Mac App Store 下载并予以安装。
  - b. 确保解压后的 Cube Demo 文件夹与安装的 Wwise SDK macOS 分发包共用同一父文件夹。比如，若 Wwise SDK 文件夹为 `${HOME}/Wwise/wwise_${version_tag}/SDK`，则 Cube Demo 根文件夹须为 `${HOME}/Wwise/wwise_${version_tag}/CubeDemo`。

- c. 在 Xcode 中，打开以下 Xcode 工程：/path/to/CubeDemo/cube\_source/src/Mac/CubeMac.xcodeproj。
- d. 构建并运行工程。

## 在 Windows 上

- 1. 方法 1：使用 BAT 脚本。
  - a. 在通过 Audiokinetic Launcher 安装 Cube Demo 后，会在 <WwiseRoot>\Cube\cube 文件夹中创建 BAT 文件。在 File Explorer 中打开此文件夹。
  - b. 双击与要运行的 Cube Demo 版本对应的 BAT 文件：Debug、Profile 或 Release 模式。或者，打开 Command Prompt 窗口，然后键入 BAT 文件的路径并按下 **Enter**。
- 2. 方法 2：通过 Command Prompt 运行可执行文件。
  - a. 找到并运行 cube.exe 可执行文件。该文件可在 <WwiseRoot>\Cube\Win32\<mode>\bin 下找到。其中，<mode> 对应于相应的模式文件夹（Debug、Profile 或 Release）。



---

## Chapter 3. 了解如何将声音引擎集成到《Cube》中

在将 Wwise 声音引擎集成到《Cube》中时涉及以下任务：

- 为 Wwise 准备《Cube》
- 注册游戏对象
- 整合音频
- 添加新的脚本命令

### 为 Wwise 准备《Cube》

《Cube》最初由 Wouter van Oortmerssen 开发，是一款开源的第一人称单人或多人射击游戏。您可以随时从 Cube 网站下载 (<http://cube.sourceforge.net/>)。为了满足 Wwise 集成包的相应需求，我们对从 Cube 网站下载的版本做了以下修改：

- 为使用 Wwise 声音引擎，改写了现有声音代码。
- 将音频内容移到了 Wwise 工程中。
- 移除了对其他库（SDL、SDL\_image、libpng）的依赖。
- 添加了额外的音频（如脚步声）。
- 为了顺应以上更改，对有些游戏配置和内容文件做了修改。

### 注册游戏对象

声音引擎中注册了两种游戏对象：

- 指向 baseent 结构体的指针。这些游戏对象是游戏世界中的移动实体：玩家、怪兽和投掷物。
- 虚拟游戏对象（编号为 64-127）。这些游戏对象用于不与移动实体绑定的 3D 声音（如物品生成）。

### 整合音频

音频整合的核心代码存放在 <WwiseRoot>\Cube\cube\_source\src 文件夹中的以下文件中：

- Sound.cpp - 包含所有访问声音引擎的代码。
- Protos.h - 包含声明的公共函数。

整合过程涉及以下类型的声音：

- 怪兽声音：
  - 每个角色都有关联的疼痛、死亡和脚步声 Event。详情请参见 monster.cpp 中的 monstertypes[]。
  - 脚步声 Event 通过 monsterfootstep() 函数发送。另外，其还会设置与角色脚下地面材质对应的 "Material" Switch 以驱动 Switch Container。
  - 物品有关联的拾获 Event。详情请参见 entities.cpp 中的 itemstats[]。
- 魔法音效：

- 每种魔法都有关联的 Event。详情参见 `weapon.cpp` 中的 `guns[]`。
- 投掷物有飞溅 Event (`S_FEXPLODE`、`S_RLHIT`)。
- 在启用四倍伤害时投掷 Ice Gem 还会发送 "S\_ITEMPUP" Event。
- 网络声音：
  - 有些 Event 还要转到服务器以便通过网络播放声音（由 `snd_clientevent()` 处理）。
- 其他声音：
  - 若要查找游戏当中发生的所有其他声音 Event，请在代码库中搜索 `snd_event()` 调用。

## 添加新的脚本命令

为方便管理声音，新增了以下与声音相关的命令。这些命令可用在 `.cfg` 脚本文件中：

- `akevent <string>`：在本地玩家游戏对象 (`player1`) 上按名称发送 Event。
- `soundvol <int>`：设置音效音量 (0-255)。
- `musicvol <int>`：设置音乐音量 (0-255)。
- `voicevol <int>`：设置语音音量 (0-255)。
- `texturematerial <int> <string1> <string2>`：该命令与 `texture` 命令基本上是一样的，不过添加了一个参数（与脚步声关联的材质）。

---

## Chapter 4. 了解 Wwise 工程

Wwise 工程方便将 Wwise 声音引擎集成到《Cube》中。为此，Audiokinetic 对原版《Cube》中的大部分声音都做了修改。



### Note

Wwise 工程文件存放在以下文件夹中: <WwiseRoot>\Cube\WwiseProject。

以下章节阐释了在 Wwise 工程中所做的一些设计决策。

### 构建 Actor-Mixer Hierarchy

您可以通过 Actor-Mixer Hierarchy 来在工程中组织声音素材。

对于《Cube》工程，我们创建了以下音频结构：

- 在 Main 文件夹中，创建了四个 Actor-Mixer 以将声音分为不同的逻辑类别：Item、Magic、Main Character 和 Monster。
- 在 Actor-Mixer 内，使用了一系列 Random Container 和 Switch Container 来让工程中的声音动态多变。
- 为部分魔法音效和脚步声的 Volume 和 Pitch 属性应用了 Randomizer。Randomizer 可在每次播放对象时随机修改其属性值。
- Main 文件夹内的所有声音都是 3D 声音。它们使用游戏定义的定位和基于标准距离的音量衰减。
- 在 Maps 文件夹中，有与 dcp\_the\_core 地图相关的特定音效、语音和音乐。此地图内的所有语音和音乐都是通过硬盘进行流播放的 2D 声音。

### 构建 Master-Mixer Hierarchy

Master-Mixer Hierarchy 是个独立的总线和辅助总线层级结构。藉此，可对工程内的诸多音效和音乐结构进行分组并为输出做好准备。

对于《Cube》工程，我们创建了以下总线结构：

- 创建了三条主要控制总线：Music、SFX 和 Voice。
- Environments 总线包含一系列辅助总线。每条辅助总线都插入有混响效果器并以此表示游戏中的房间。由游戏代码决定在运行时处理哪条辅助总线。
- 向 Voice 总线应用了 Auto-ducking 以在播放 Voice 时闪避 Music。
- 向三条主要控制总线的 Volume 属性应用了 RTPC。这些音量控件被映射到游戏内的音量推子（通过游戏主机）。

### 使用 Game Sync

在 Wwise 中，可利用 Game Sync 高效地管理与游戏内的动作或条件变化相关的特定音量变化。在《Cube》工程中，使用了以下 Game Sync：

- **Switch** – 创建了 "Material" Switch Group 以管理游戏中存在的不同地面材质。在此 Switch Group 内为各种地面材质（如水泥和草地）创建了 Switch。在游戏内的地面发生变化时，可使用这些 Switch 来为主角定义不同的脚步声。
- **Game Parameter** – 创建了三个与《Cube》游戏内的音量推子对应的 Game Parameter。这些 Game Parameter 通过 RTPC 映射到三条主要总线的 Bus Volume 属性。通过将推子映射到音量属性，游戏玩家可在游戏中自行控制音效和音乐的音量。

"PlayerHealth" Game Parameter 与玩家在游戏中的 HP 值绑定。在默认情况下，没有声音使用此 Game Parameter。您可以将此 Game Parameter 绑定到声音对象的属性。

Teleport 循环 Synth One 声音在合成器的 Base Frequency 属性上使用 "Distance\_to\_Object" Game Parameter 来在靠近传送门时生成有趣的弯音效果。此 Game Parameter 可重复用在其他声音对象上。

## 创建 Event

Wwise 使用 Event 驱动游戏中的音效、音乐和对白。Event 包含一个或多个应用于工程层级结构内的不同音效或音乐结构的动作。对于《Cube》工程，我们创建 Event 并为其采用了与《Cube》触发的声音匹配的名称。

## 了解 SoundBank

SoundBank 涵盖工程内的所有成果。这些最终的音频包会成为游戏的一部分。它们包含一组 Event、Wwise 对象和/或转码音频文件，并会在游戏中的特定时刻一起加载到游戏平台内存中。对于《Cube》工程，我们创建了以下两个 SoundBank：

- **Main** – 包含所有地图中使用的全部 Event。
- **dcp\_the\_core** – 包含 dcp\_the\_core 地图中使用的 Event。

---

## Chapter 5. 调整声音设计

以下章节阐释了如何实时微调音频并将这些更改整合到游戏中。

### 在游戏中实时微调音频

在连接到《Cube》游戏后，可通过在游戏当中实时修改各个声音的属性来微调音频。



#### Note

如需了解如何将 Wwise 设计工具连接到游戏，请参阅 Wwise Help 中的[连接至本地/远程游戏系统](#)章节。

在将 Wwise 连接到游戏后实时修改声音属性：

1. 在《Cube》中，左键单击来投掷 Ice Gem 并试听当前声音。
2. 在 Wwise 中，选中 "IceGem\_Blast" Sound SFX 来将其加载到 Property Editor 和 Transport Control 中。

"IceGem\_Blast" Sound SFX 设在 Main 文件夹中的 "Magic" Actor-Mixer 下。

3. 修改 "IceGem\_Blast" Sound SFX 的 Volume 和 Pitch 属性。
4. 单击 Play 试听新的声音。
5. 返回《Cube》并再次投掷 Ice Gem。

这时会在游戏中自动播放新的声音。

6. 试着对游戏中的其他声音执行类似的操作，体验一下如何快速、轻松地微调游戏音频。

### 将更改整合到游戏中

虽然可在连接到游戏时试听差异，但是所有这些修改都只是暂时的。若要将这些更改整合到游戏中，则须重新生成并替换现有 SoundBank。在替换 SoundBank 后，会将新的声音整合到《Cube》游戏中。



#### Note

在生成新的 SoundBank 前确保关闭《Cube》。在游戏运行时，游戏可能会锁定一个或多个 SoundBank，确保无法正常覆盖磁盘上的 SoundBank。

为《Cube》生成新的 SoundBank：

1. 通过执行以下任一操作来切换到 SoundBank 布局：
  - 在菜单栏中，依次单击 Layouts > SoundBank。
  - 按下 F7。

这时会为此工程生成两个 SoundBank：DCP\_the\_core 和 Main。

2. 在 SoundBanks 列表中，单击 Select All。

3. 在 Platforms 列表中，根据所用操作系统选中 Windows 或 Mac 选项。
4. 在 Languages 列表中，选中 English (US) 选项。
5. 在菜单栏中，依次单击 Project > Project Settings。

这时会打开 Project Settings 对话框。

6. 切换到 SoundBanks 选项卡。
7. 在 Post-Generation Step 分组框中，确保有将工程中的流媒体文件复制到 SoundBank 保存位置的命令行。在默认情况下，所有工程都将此命令行作为生成后步骤。
8. 单击 OK 关闭 Project Settings 对话框。
9. 单击 **Generate Checked** 开始生成 SoundBank。

这时会打开 Generating SoundBanks 对话框。在此，可查看 SoundBank 生成过程的进度。在生成 SoundBank 后，会打开 SoundBanks Generation - Completed 对话框。

10. 单击 Close。

这时会生成新的 SoundBank 并将其集成到《Cube》游戏中。在下次试玩《Cube》时，游戏中会包含新的声音。

## 已知问题

鉴于该示例游戏只是为了展示如何将 Wwise 声音引擎集成到游戏中，此版本中并未解决以下问题。

### 已知 SDL 迁移问题

- 无法全屏显示。
- 无法使用 Gamma 命令。
- 无法在运行专用程序时查看服务器消息。
- 帧率较低时游戏中的鼠标光标会消失。
- 无法通过鼠标滚轮选择武器。

### 已知声音问题

- 被遮蔽的角色不会发出脚步声。
- 在水中也会听到脚步声。
- 没有互动音乐。

---

## Chapter 6. 需要帮助?

### 使用 Help

Wwise Help 包含 Wwise 中各项界面元素的详细信息。

若要在 Wwise 内打开帮助文档，请执行以下任一操作：

- 单击任意视图或对话框标题栏中的 Help 图标 (?)。
- 在菜单栏中，依次单击 **Help > Wwise Help**。
- 按下 F1。

### 其他资源

《Cube》游戏及相应 Wwise 工程方便了解如何使用 Wwise 开发一个完整的游戏。若要深入了解如何使用 Wwise 来以不同的方式整合游戏音频，建议参考以下资源：

- 《Limbo》 - 可通过 Audiokinetic Launcher 随 Wwise 工程一并安装该游戏。  
《Limbo》是由 Playdead 开发的一款屡获殊荣的游戏。《Limbo》比较适合高级 Wwise 用户，因为其包含一些复杂的声音层级结构。注意，无法在 Mac 上通过 Audiokinetic Launcher 安装《Limbo》。
- 视频教程 - 可通过官网上的 [Audiokinetic 视频频道](#)和 [Audiokinetic 的 YouTube 频道](#)观看视频教程。
- Wwise 认证 - [Wwise 认证](#)在线课程提供各种用来学习 Wwise 基础知识和进阶课题的培训资料。注意，Wwise-101 和 Wwise-201 认证计划使用《Cube》作为开展 Wwise 教学的首选平台。