WWise

Events

Events（事件）由 Wwise 用户创建，方法是在 Wwise 对象中指定要执行的动作。

然后把事件打包成 SoundBank ，以便加载到游戏中，此后，游戏代码便可触发这些事件。

事件一旦集成到游戏中，Wwise 用户就可以继续打磨这些事件，更换或修改事件包含的动作或者它们引用的对象。由于游戏触发的仍是同一事件，因此不再需要程序员动手介入，也无需重新编译代码，Wwise 用户所做的更改就会在游戏中生效。

SoundBank

SoundBank 是包含游戏音频数据的文件。它们可由 Wwise 用户在 SoundBank Manager（SoundBank 管理器）的帮助下生成。也可通过自动化过程定义并生成。

有两种类型的 SoundBank ：初始化 Bank和SoundBank。

每个工程只有一个初始化 Bank，名称为“Init.bnk”。启动游戏时加载的第一个 SoundBank 必须是初始化 Bank。

SoundBank 包含 Wwise 事件以及播放它们所需的所有对象和音频数据。

您的游戏必须首先加载初始化 Bank。然后再根据需要加载/卸载 SoundBank。具体加载什么 SoundBank 将取决于在游戏中某点可能触发的事件。

Game Object

一个 Game Object（游戏对象）通常代表一个游戏元素。对于可以发出声音的各个游戏内元素，需要通过 AK::SoundEngine::RegisterGameObj() 来给它们注册 Game Object。

Switch

在游戏的某些场景，您可能希望根据游戏的当前状况播放不同的声音。举例来说，假如角色先在混凝土地面上奔跑，然后进入草地，脚步声应随着地面的变化而变化。要在不同声音之间进行选择，应该使用 Switch（切换开关）。

通过使用 Switch 和 Switch Container（切换容器），Wwise 用户可定义每种情境中播放的声音。在 Wwise 中可创建多个 Switch Group（切换开关组）来匹配您可能在多种声音之间进行选择的情形。每组可以包含多个 Swtich。于是 Switch Container 可以使用这些 Switch Group 并将声音与各个 Switch 关联起来。

State

当游戏中的条件发生变化时，例如角色潜入水中时，您希望声音能对发生的情况进行响应。这意味着任何枪声、背景音乐或角色的动作听起来应该像角色正在水中一样。要使音频与当前游戏情境匹配起来，应该使用 State（状态）。

在 Wwise 中可创建不同的 State Group（状态组）来代表不同类型的游戏状态，每个 State Group 可包含多个 State。例如，游戏中可以包含以下 State Group 和 State，它们由声音设计师在 Wwise 中定义。

实时参数控制（RTPC）

Real-time Parameter Controls（实时参数控制，RTPC）用于根据游戏中发生的实时参数变化，实时控制各种 Wwise 对象（包括音效对象、容器、总线、效果器等）的特定属性。

Listener

Listener（听者）代表游戏中的话筒位置，因此可以在扬声器中定位 3D 声音来模拟真实的 3D 环境。这是通过将 Game Object（游戏对象）的位置信息映射到 Listener 的坐标系统中来实现的。该坐标系统通过位置向量、前方朝向向量和上方朝向向量进行定义。因此必须即时更新 Listener 的位置信息，否则将通过错误的扬声器播放声音。

环境和 Game-defined Auxiliary Sends

Wwise 的 Game-defined Auxiliary Sends 便于根据游戏中声音所在的位置来为其应用若干效果器。 在对象或者其可以继承的上层对象中必须已经设置了选项“Use game-defined auxiliary sends”。

高级播放设置——播放限制和优先级

在游戏进行中的任何时刻，都可能会有同时播放的大量声音，触发的声音数量甚至会多到超出硬件的处理能力。为了有效地管理播放声音的数量，您必须决定同时可以播放多少个声音，以及优先播放哪些声音。

Wwise 中有两个主要属性可以帮助您确定将在游戏中播放的声音：

Playback Limit——播放限制。指定任意时刻可以播放的声音数量限制。

Playback priority——播放优先级。指定声音对象之间的相对重要性。

另外，您可以决定设置一些声音在超出限制时不被终止播放，而是

采取虚行为

或者

超出限制时继续播放。当太多声音的虚行为被设置为“Continue to play”时，最后的情况可能是超出限制。

Virtual Voice

为了在大量声音同时播放时保持最佳的性能水平，低于特定音量电平的声音不应占用宝贵的处理器机能和内存。声音引擎可以不播放这些无法听到的声音，而将它们放到虚声部列表中排队。Wwise 可以继续管理和监控这些声音，但是这些声音一旦进入虚声部列表，就不再由声音引擎处理，因而不会占用硬件的任何一个实声部。

当您选择使用虚声部功能时，声音可以在 physical voice（实声部）和 virtual voice（虚声部）之间来回切换，方法是指定当声音的音量电平低于阈值或声音数超出播放限制时，允许让这些声音进入虚声部。当音量达到 Wwise 用户在 Project Settings（工程设置）中设定的阈值时，声音将被添加到虚声部列表，并且声音处理工作停止。随着音量电平的上升（当声音从外部移入到最大距离半径以内时就会发生这种情况），声音将从虚声部列表移到实声部，在此声音将再次由声音引擎进行处理。

Trigger

Trigger（触发器）是让您启动被称为 Stringer（插播乐句）的游戏同步器，Stringer 是根据具体情况自发出现的乐句。这些乐句响应游戏中发生的事件，叠加在当前正在播放的音乐上，并与之混合。触发器的同步机制可以利用已有音乐的节奏使 Stinger 的集成变得天衣无缝。

例如，当前正在播放的音乐可以响应于游戏触发器“Main Player Hit Bullseye”，在保持当前节奏的同时在当前音乐上播放鼓加花。这一鼓加花被称为 Stinger。

第一次安装 Wwise 集成包

集成过程由 Wwise Launcher 负责。请使用 Wwise Launcher 来将 Wwise 集成到 Unity 工程中。