## （一）类或组件说明

（1）**AudioInfo类**：称音频信息类，用于记录音频信息，其区别于AudioSource仅记录常用的音频属性，但是通过该类也可以获取AudioSource组件，该类提供了通过AudioSource组件记录并返回信息、播放音频、暂停音频播放、绑定音频管理器、初始化为默认值、将信息配置到指定AudioSource组件以及将指定AudioSource组件的信息记录到该类等方法。

（2）**AudioManager组件**：称音频管理器组件，用于管理音频信息，其包括是否进行音频信息覆盖、音频信息数量、总音量更改回调事件和音频信息名称合集等属性，除此之外还提供了播放指定音频、播放音频并开启立体声过渡、暂停播放指定音频、释放音频资源以及添加和删除音频信息等方法。

（3）**AudioSourcePool类**：称AudioSource组件池类， 用于统一管理AudioSource组件，其提供了获取和归还AudioSource组件等方法。

## （二）使用步骤

1.在游戏对象上挂载AudioManager组件；

2.在Inspector面板设置相应的属性，如图1所示：



图 1 AudioManager组件

3.属性介绍：

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **取值范围** | **特殊说明** |
| TheAudioClip | component | AudioClip | 无 |
| AudioName | string | 无 | 无 |
| Volume | float | [0,1] | 默认为1 |
| Pitch | float | [-3,3] | 默认为1 |
| StereoPan | float | [-1,1] | 默认为0 |
| Priority | int | [0,256] | 默认为128 |
| PlayOnAwake | bool | true or false | 默认为true |
| Loop | bool | true or false | 默认为false |
| IgnoreTotalVolume | bool | true or false | 默认为false |
| StereoTransition | bool | true or false | 默认为false |
| StereoTransitionTimeSpan | float | [0.1,5] | 默认为0.5 |
| StereoTransitionValues | float[] | 无 | 无 |

|  |  |
| --- | --- |
| **名称** | **说明** |
| TheAudioClip | AudioClip组件 |
| AudioName | 音频名称 |
| Volume | 音频音量 |
| Pitch | 音频播放速度 |
| StereoPan | 立体声设置 |
| Priority | 音频优先级 |
| PlayOnAwake | 是否在场景启动时进行播放 |
| Loop | 是否循环播放 |
| IgnoreTotalVolume | 是否忽略总音量的影响 |
| StereoTransition | 是否启用立体声过渡 |
| StereoTransitionTimeSpan | 立体声过渡的每帧时间间隔 |
| StereoTransitionValues | 立体声过渡值集合 |

4.通过代码调用开始和暂停音频播放以及开启立体声过渡效果，代码如下：

**TheAudioManager.Play("BGM1");//直接根据名称按照在Inspector面板预设好的属性播放音频**

**float stTimeSpan = 0.1f;**

**float[] stValues = NumberRange.FloatRange(-1, 1, 0.01f, true);//通过NumberRange中的FloatRange生成立体声过渡值集合**

**TheAudioManager.PlayWithStereoTransition("BGM1", stValues, stTimeSpan);//根据名称播放音频并开启立体声过渡**

**TheAudioManager.Pause("BGM1");//暂停播放"BGM1"，暂停不会释放占用资源**

**TheAudioManager.Dispose("BGM1");//释放"BGM1"所占用的资源**

## （三）API

*请查看同目录下的AudioManagerAPI.xmind文件…*

## （四）F&Q

**Q1：AudioManger有哪些优点？**

A1：AudioManager开发的初衷则是为了统一管理音频，为音频的操作提供便利，原有的方式则是在需要使用音频的游戏对象上挂载AudioSource组件，然后设置对应的音频属性，然后通过代码获取AudioSource组件控制音频播放。而AudioManager组件则作为一个音频信息库，可以提前在其中添加待使用的音频信息，将音频信息集中管理，并且它将提供统一的接口对音频进行操作，它的内部会结合AudioSourcePool自动为待播放的音频分配AudioSource组件，以此减少资源开销。除此之外AudioInfo也可以作为可序列化的音频信息实体类进行存储，虽然AudioSource也可以序列化，但是AudioSource的体量更大，通常有一些音频信息我们并不需要去存储，所以AudioInfo主要是提取AudioSource中常用的音频信息。

**Q2：立体声过渡是什么？**

A2：立体声过渡是基于AudioSource的StereoPan属性实现的一种左右声道过渡的效果，通过协程动态改变StereoPan从而实现立体声过渡，后续还会有其它音效的实现。