

Document Explicatif

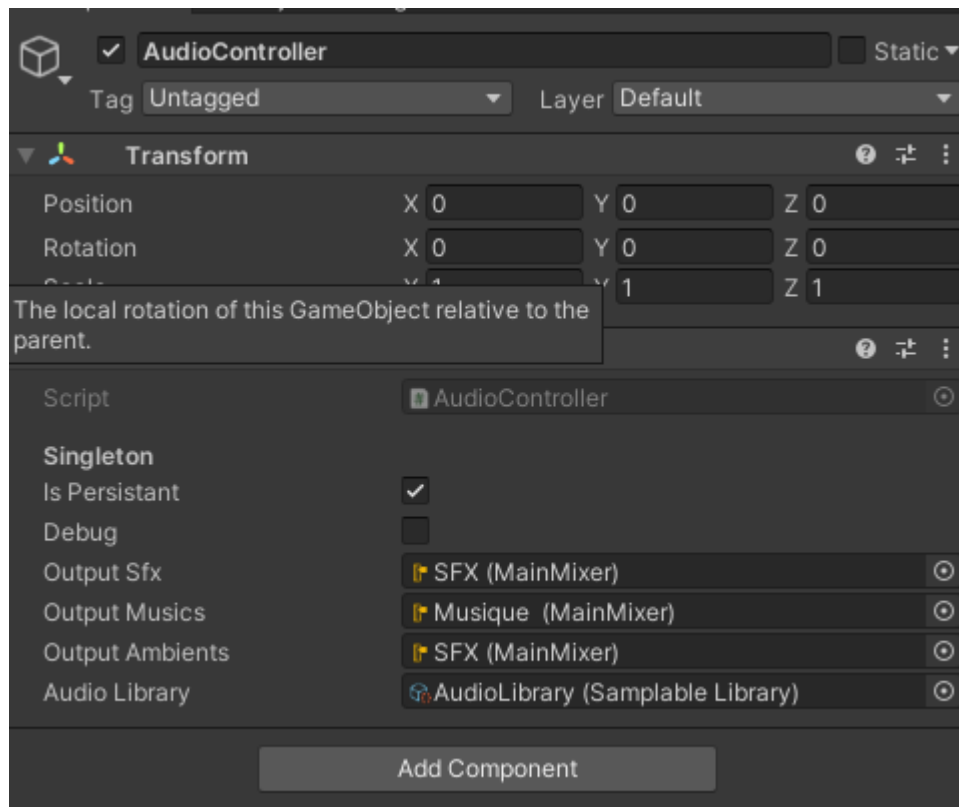
GESTION DU SON

Sommaire :

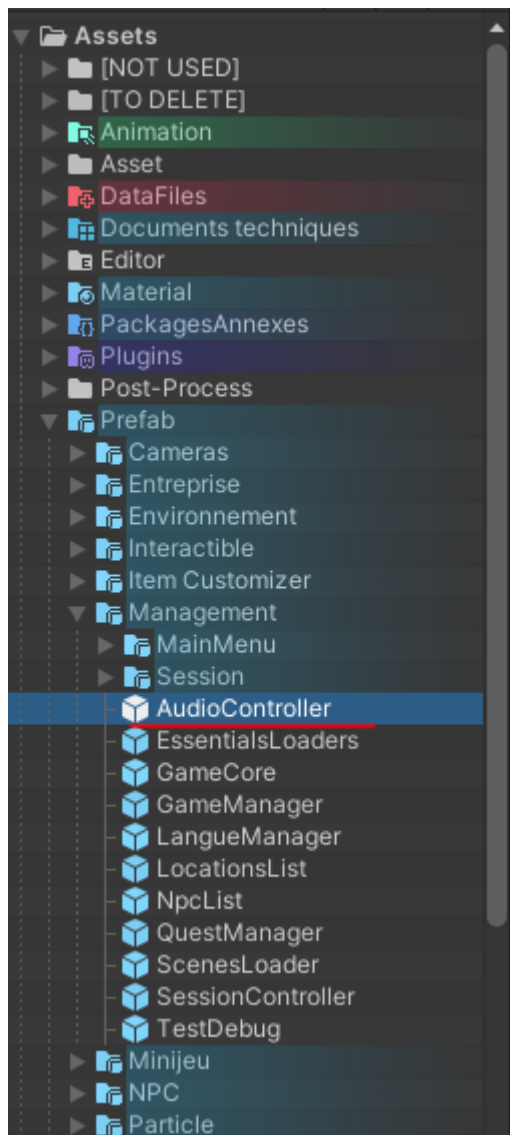
- 1) Gestion du son
- 2) Création des audio datas
- 3) Répertoires des principaux scripts utilisés

1) Gestion du son

La gestion du son dans le jeu fonctionne à partir du script **AudioController.cs**, un **singleton** à partir duquel on va appeler les sons à jouer, et qui stocke les différents **audio mixers** contrôlant le volume des sons selon leur type (**SFX**, **Music**, **Ambiance**, etc...) :



Le prefab stockant ce script et qui doit être présent en permanence sur la scène pour pouvoir utiliser des sons, se trouve au chemin : **Prefab > Management** :



Ensuite, on peut appeler ce script via différentes méthodes pour manipuler les sons (pour plus de précisions, voir directement dans le script **AudioController.cs**) :

```
AudioController.Instance.PlayAudio(sfxAddQuest);
```

La méthode **PlayAudio** permet de simplement lancer la lecture d'un son donné en paramètre.

```
AudioData[] datas = GetRandomClip();  
AudioController.Instance.PlayRandomOneShotAudio(datas);
```

La méthode **PlayRandomAudio** permet de lancer la lecture d'un son aléatoire parmi un tableau de sons passé en paramètre.

```
AudioController.Instance.StopAudio(GameCore.Instance.MainMusic);
```

La méthode **StopAudio** permet de stopper un son type passé en paramètre.

```
AudioController.Instance.RestartAudio(GameCore.Instance.MainMusic);
```

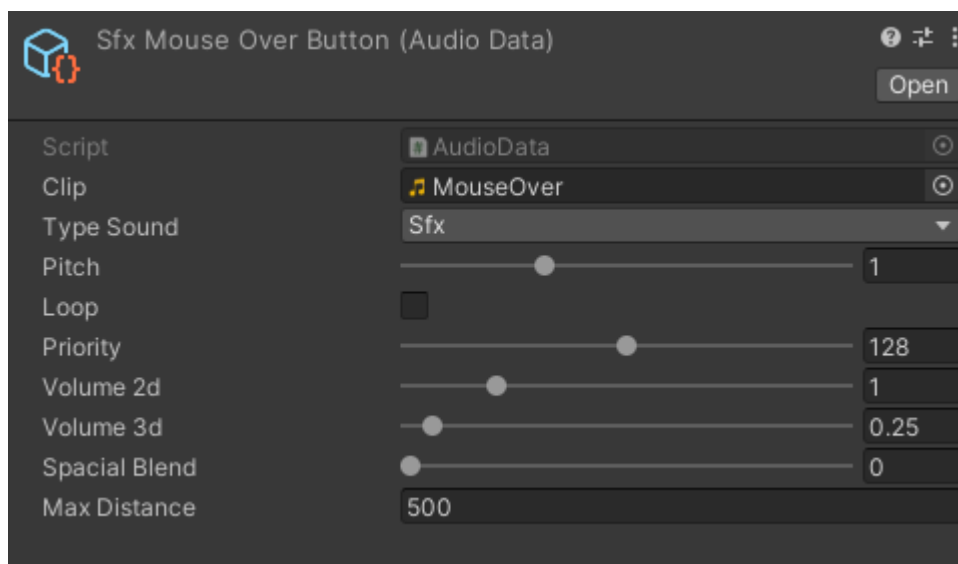
La méthode **RestartAudio** permet de stopper un son type en cours passé en paramètre, puis le relancer juste après.

```
AudioController.Instance.StopAllAmbients();  
AudioController.Instance.StopAllMusics();  
AudioController.Instance.StopAllSfx();  
...
```

Les méthodes **StopAllAmbients**, **StopAllMusics**, et **StopAllSfx**, permettent respectivement de couper tous les sons d'ambiance actifs, toutes les musiques, ou tous les Sfx.

2) Création des audio datas

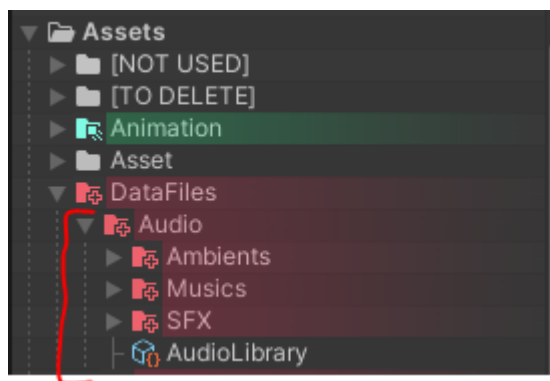
Les sons peuvent ensuite être définis et configurés grâce à des fichiers datas, les audio datas qui seront appelés par le script **AudioController.cs** pour gérer les sons, et dans lesquels on peut définir différents paramètres :



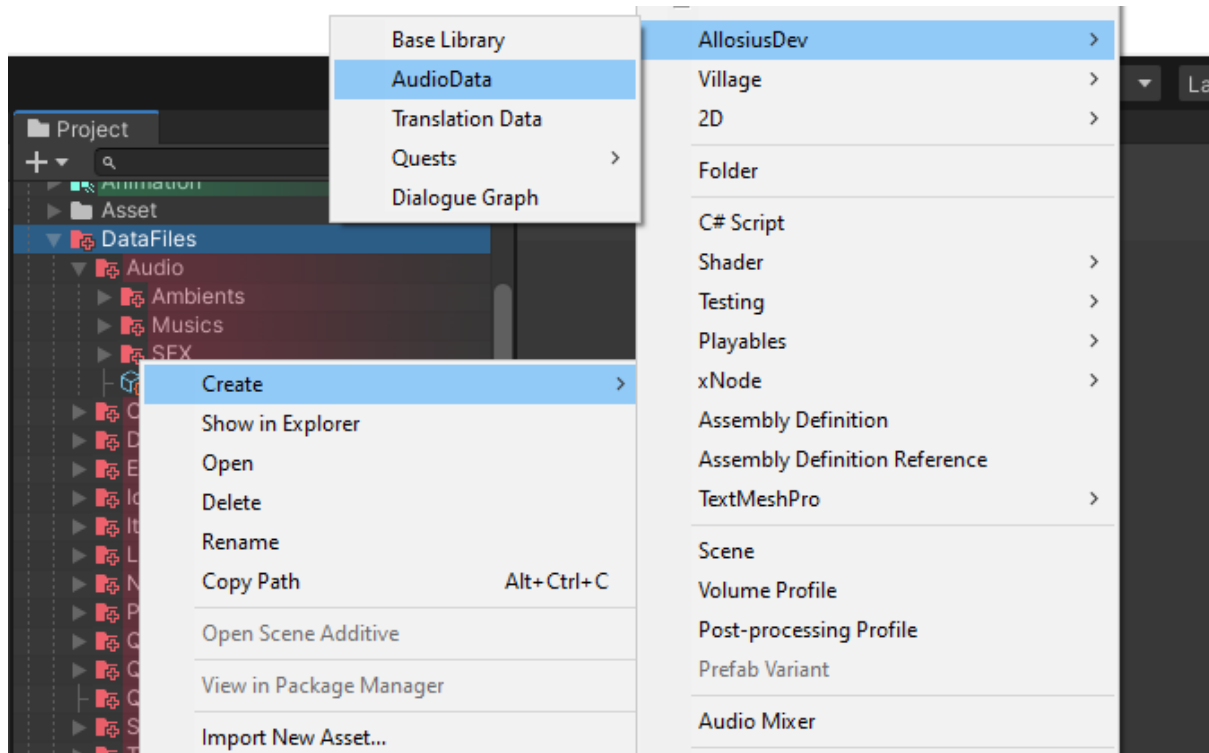
- **Clip** : clip audio du son à jouer
- **Type Sound** : type du son définissant quel **audio mixer** va contrôler son volume (**Sfx, Music, Ambient**)
- **Pitch** : définit le pitch à donner au son joué (par défaut le laisser à 1)

- **Loop** : définit si le son créé doit être joué en boucle ou non
- **Priority** : définit le niveau de priorité du son joué
- **Volume 2d** : définit le volume sonore du son (si le **spatial blend** n'est pas activé et donc que le son n'est pas joué en fonction de sa position dans l'espace)
- **Volume 3d** : définit le volume sonore du son (si le **spatial blend** est actif et que donc le son est joué en 3d)
- **Spatial Blend** : définit si le son joué va être calculé dans l'espace ou non
- **Max Distance** : définit la distance maximale entre le son et l'**audio listener** de la scène de jeu, pour laquelle celui-ci va pouvoir être entendu (si **spatial blend** actif)

Les **audio datas** sont stockés au chemin : **DataFiles > Audio** :



Pour les créer, vous pouvez le faire via le menu : **Create > AllosiusDev > AudioData** :



3) Répertoire des principaux scripts utilisés

Vous pouvez retrouver l'ensemble des scripts utilisés par le système de gestion du son au chemin : **PackagesAnnexes > AllosiusDevUtilities > Scripts** :

