

SOUND SYSTEM

IMPLEMENTATION

Germán López Gutiérrez

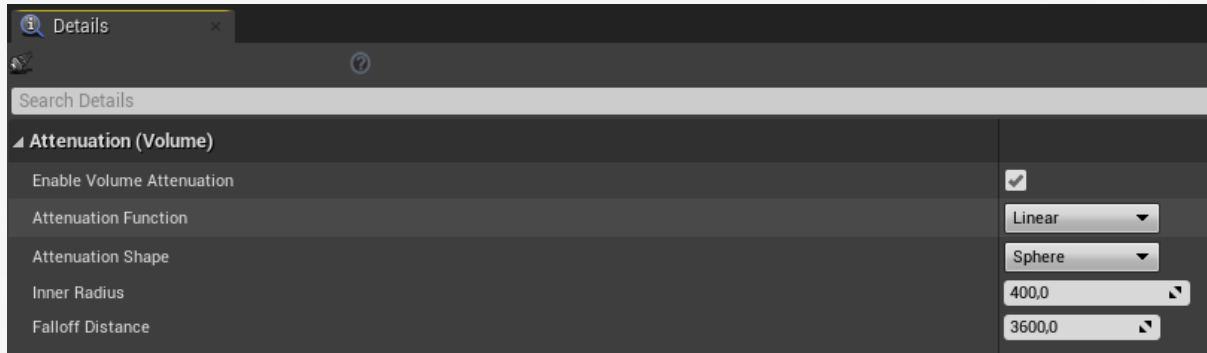
Content

Definitions.....	3
Attenuation.....	3
Concurrency.....	3
Implementation.....	4
Data Structure.....	4
Sound Mixer.....	4
Sound Class.....	4
Sound Hierarchy.....	5

Definitions

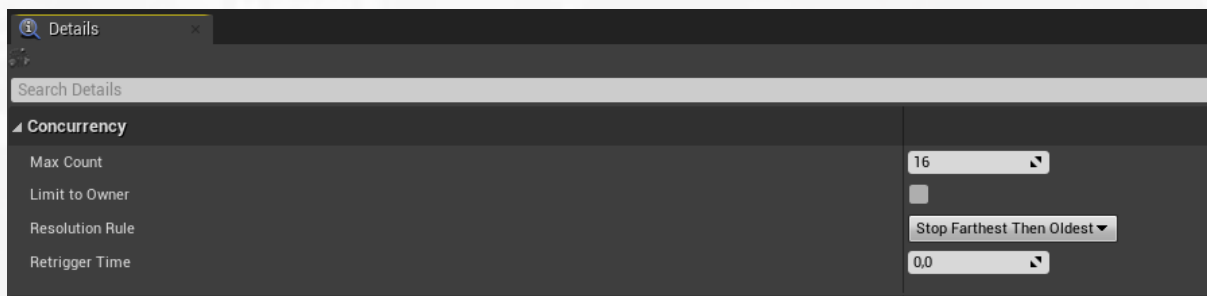
Attenuation

Manages the sound effect in terms of the 3D environment. The most used are the Logarithmic for almost all elements (enemy voices, gunshots, etc.) and the Natural for objects in the environment (a campfire for example).



Concurrency

Controls the amount of the same sound that can be played simultaneously. Ideally, there should be no overlapping sounds.



Implementation

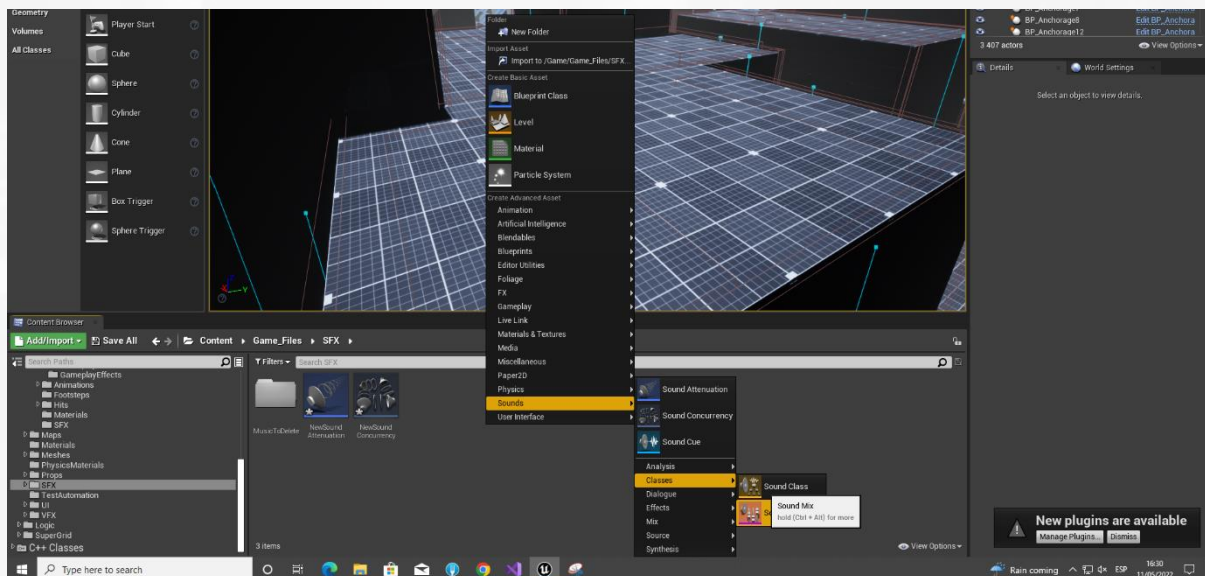
Data Structure

It is required to implement data structure composed of the following elements:

- Volume (float)
- Pitch (float)
- StartTime (float)
- Attenuation
- Concurrency
- IsUISound (bool)
- AutoDestroy (bool)

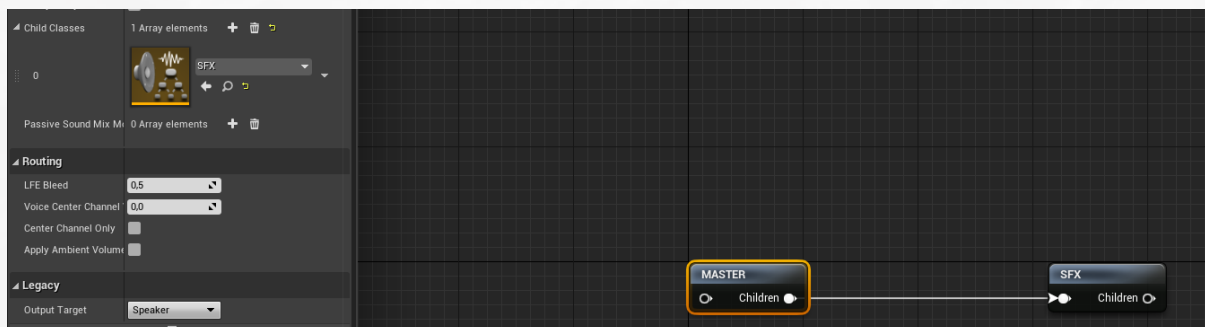
Sound Mixer

When creating the sound system, a single Sound Mixer must be generated. This is where the sound types will be managed: SFX, MUSIC, etc.



Sound Class

It is the class that contains the sound types and their inheritances (Master contains SFX, VOICE and MUSIC for example).



Sound Hierarchy

As for the data tables starting from the required data structure, they follow the following hierarchy and must be created from right to left (being the ones on the right of the whole Data Table and starting from the next Composite Data Table). Moreover, this is the same hierarchy that must be followed with respect to the Sound Classes.

