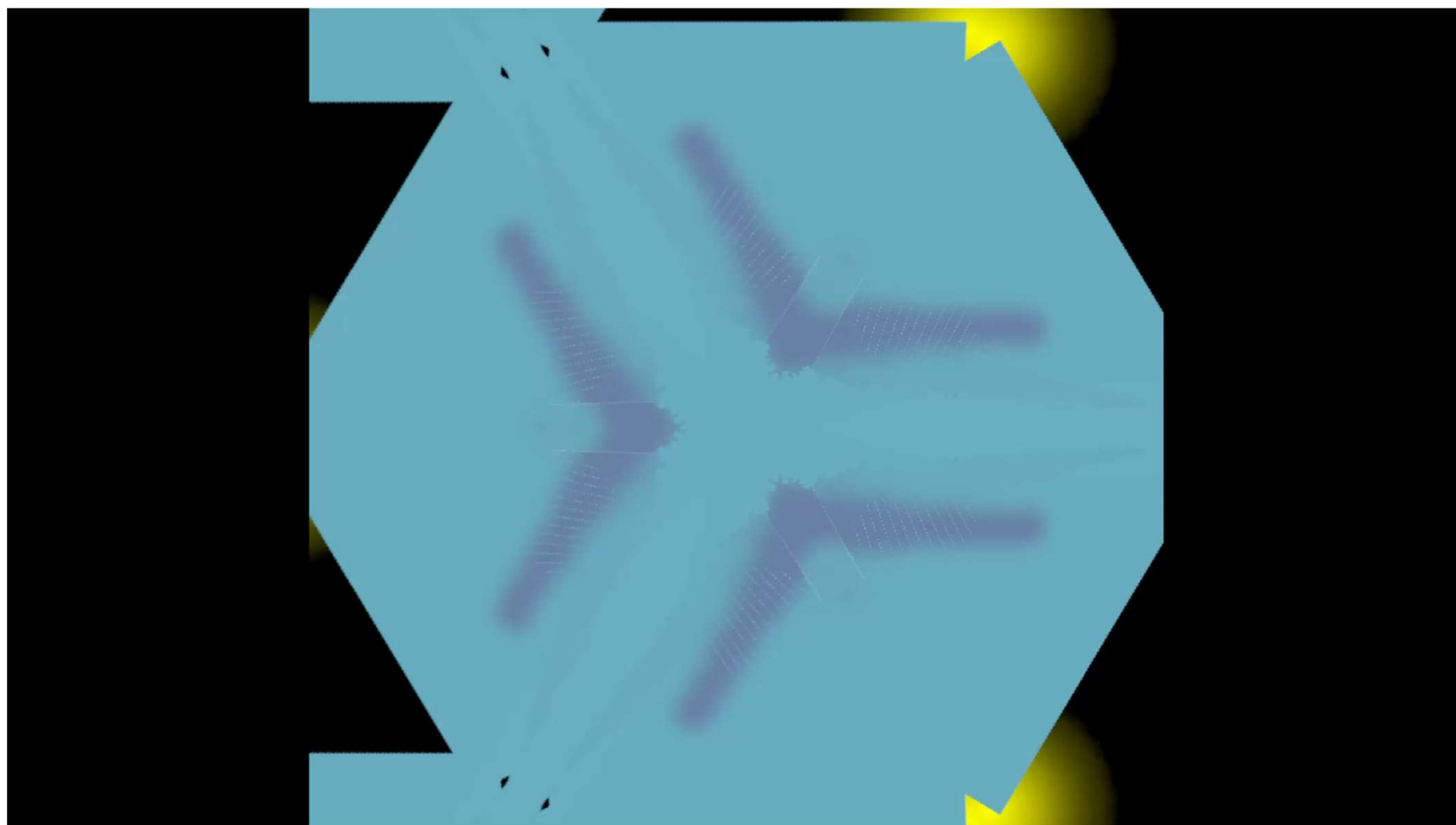
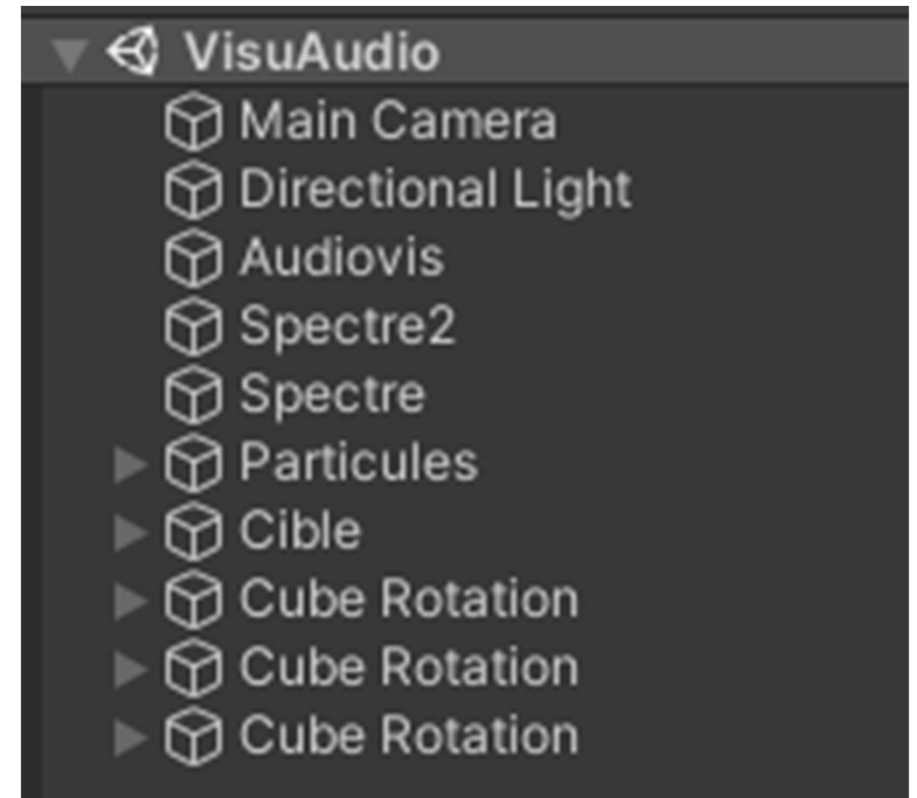


# Visualisation Audio Interactive

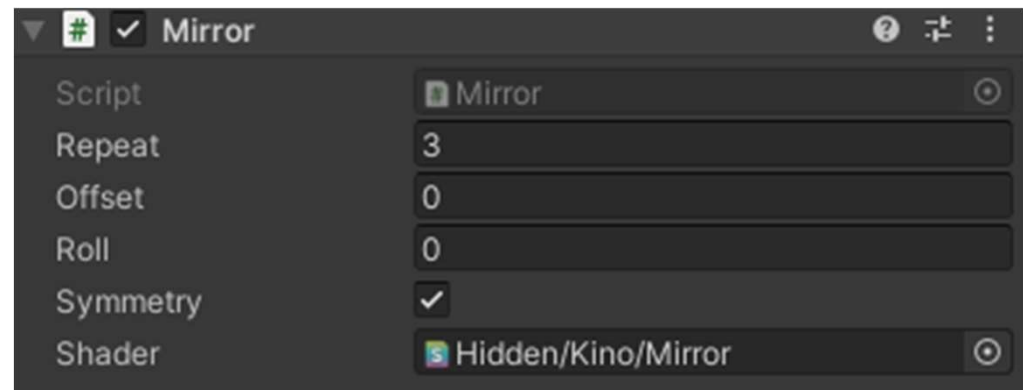


La scene s'organise ainsi



# Main Camera

Contient le script « Mirror » qui permet de paramétrer le shader tierce Kino (pour l'effet kaleidoscopique)

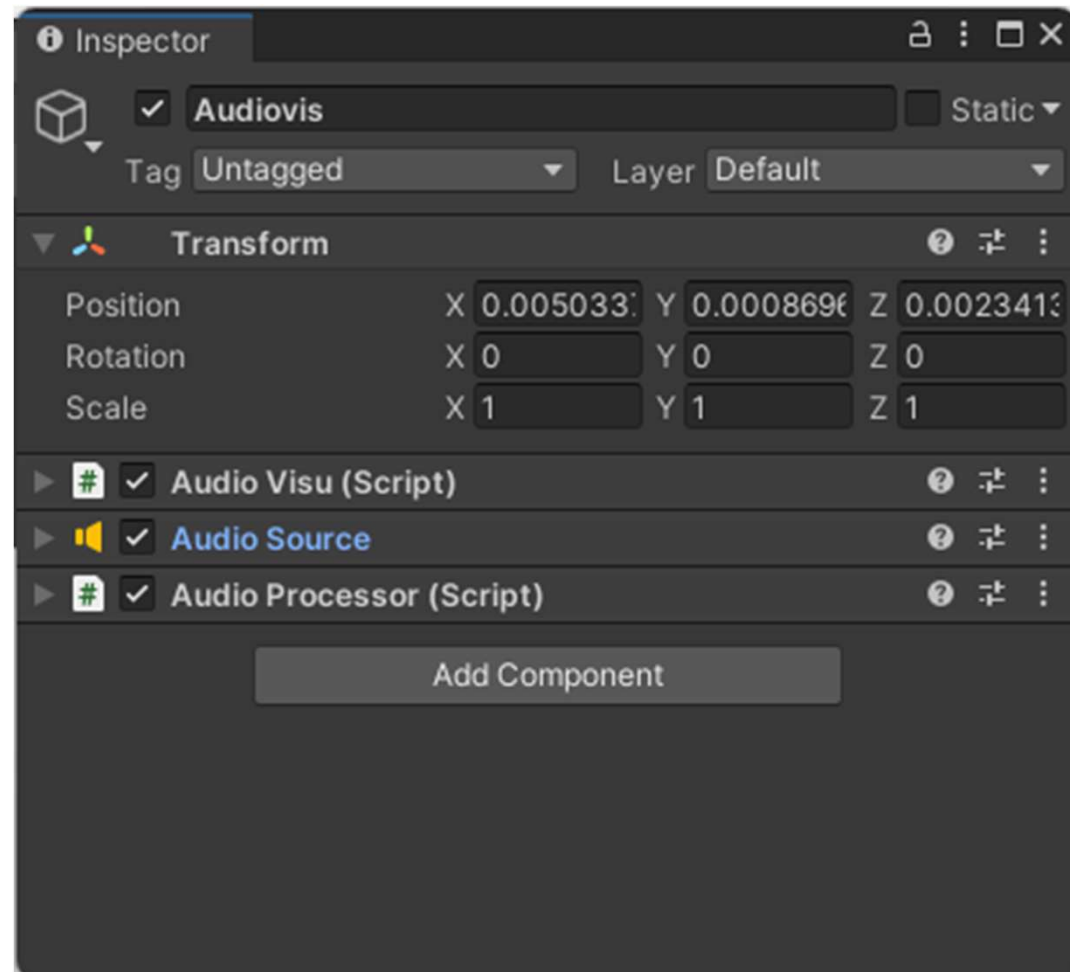


# GameObject Audiovis

Contient la source audio dans Audio Source

Contient le script AudioVisu.cs qui va récupérer le spectre audio.

AudioProcessor.cs un script tierce partie récupéré sur le web (voir le code source pour plus d'information) qui fait comme AudioVisu.cs mais avec des fonctionnalités supplémentaires

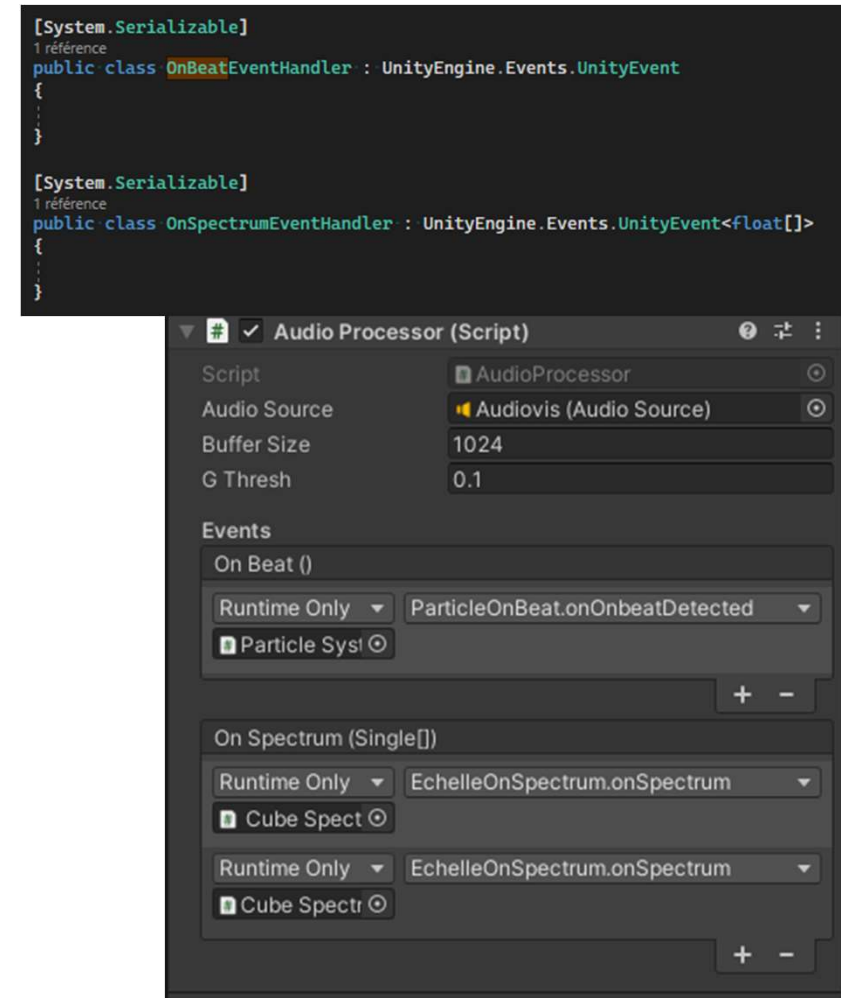


# GameObject Audiovis

AudioProcessor.cs donne accès à des événements

L'un des événements détecte le tempo (beat)

L'autre permet d'accéder au spectre audio



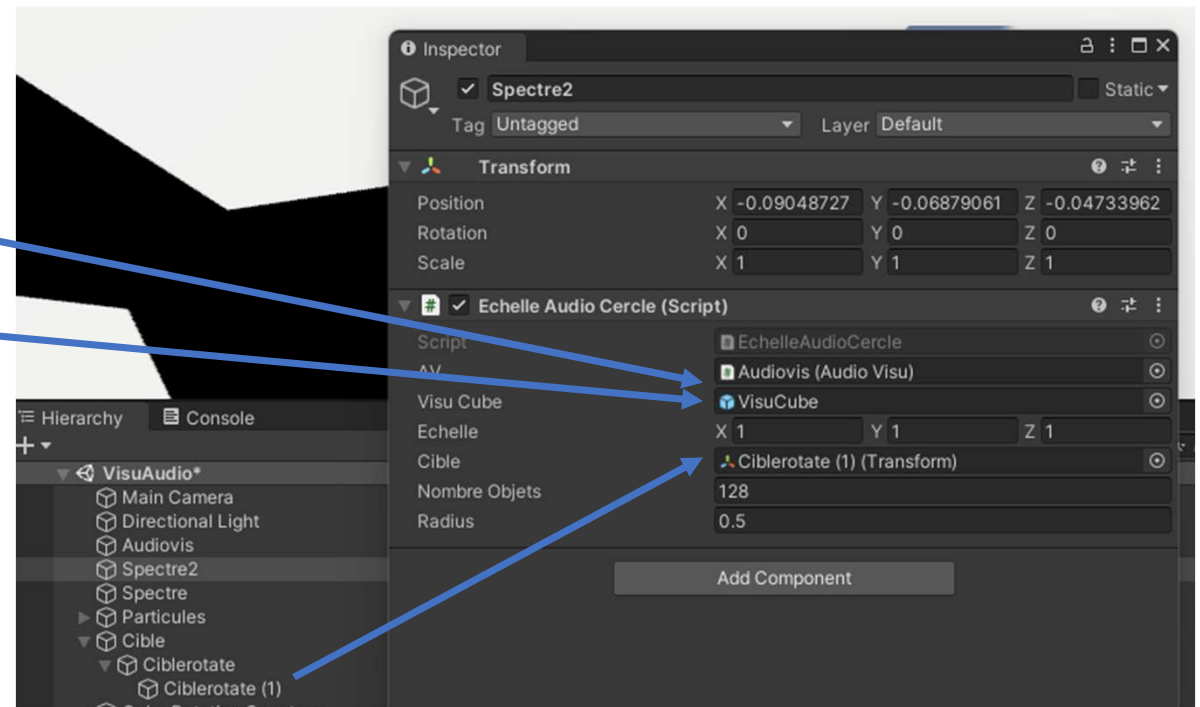
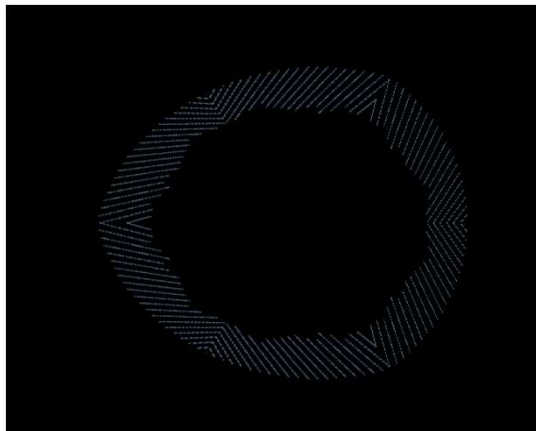
# GameObject Spectre 2

Contient le script EchelleAudioCercle.cs

Va chercher le spectre via AudioVis.cs

Utilise un prefab pour la representation du spectre (ici VisuCube)

Et une cible pour aligner/orienter les prefab instancié

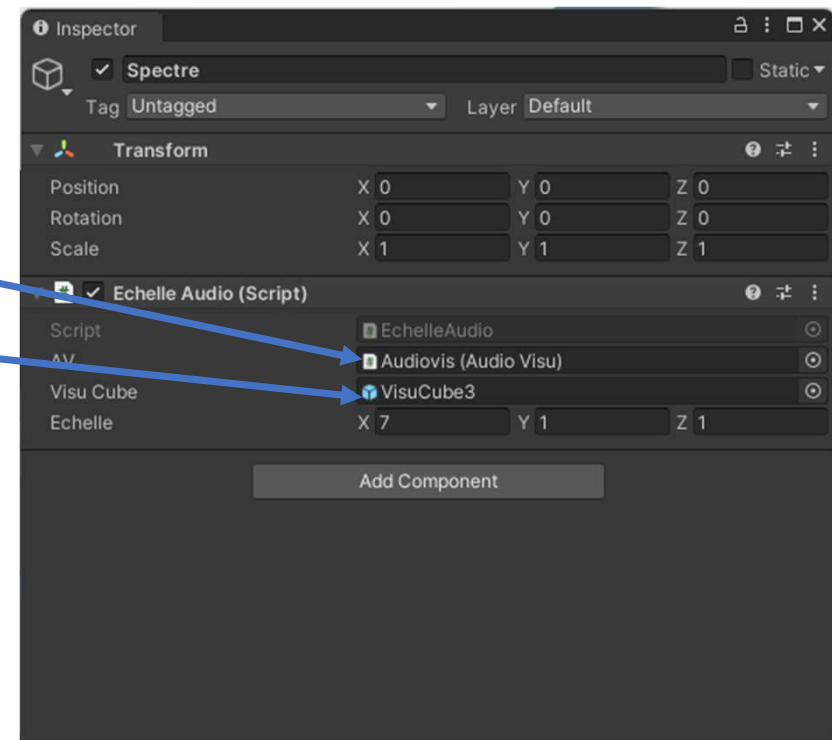
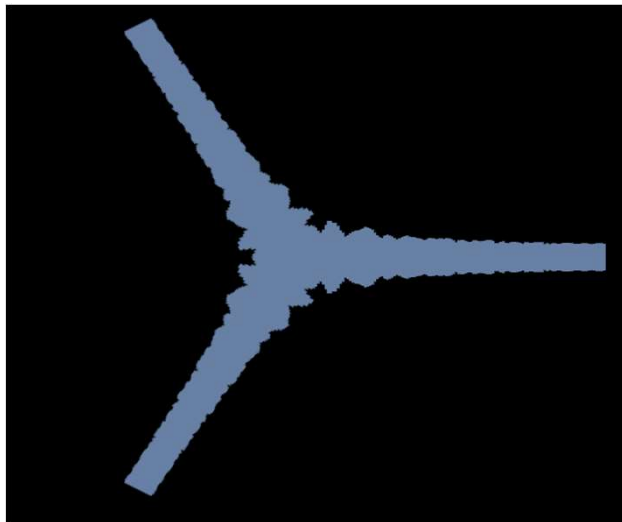


# GameObject Spectre

Contient le script EchelleAudio.cs

Va chercher le spectre via AudioVis.cs

Utilise un prefab pour la representation du spectre (ici VisuCube)





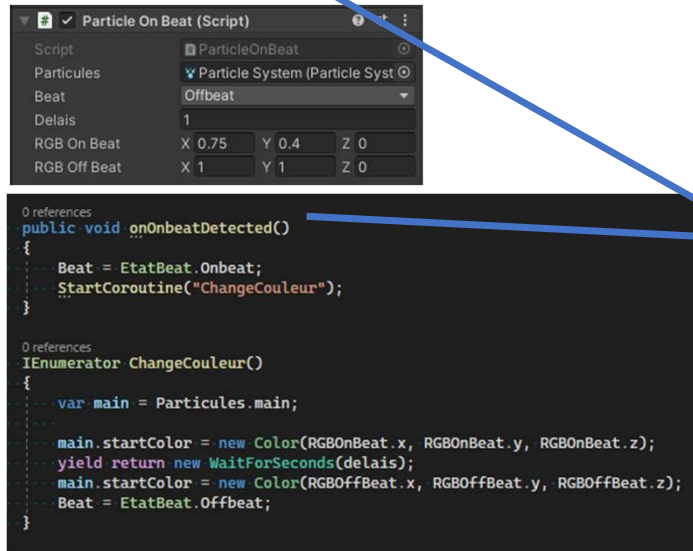
# GameObject Particules

Contient un système de particules standard de Unity

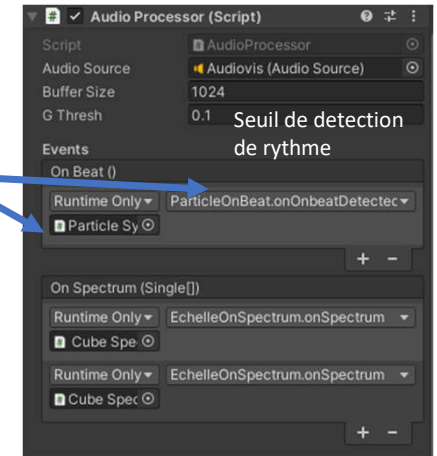
Change de couleur en fonction du rythme avec la fonction ParticleOnBeat via l'événement OnBeat de AudioVis>AudioProcessor

L'état est défini dans ControleBeat.cs

GameObject  
ParticleSystem



GameObject  
AudioVis



# GameObject CubeSpectrum

Contient un Cube dont la taille change avec la fréquence.

Ceci se fait grâce au script Echelle On Spectrum

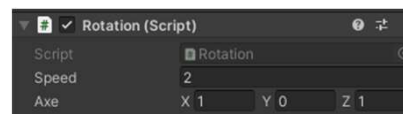
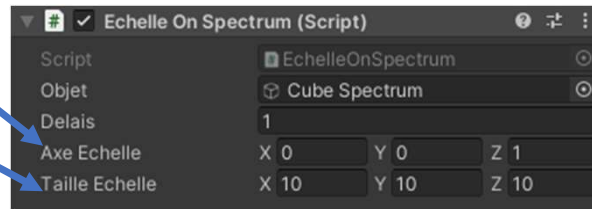
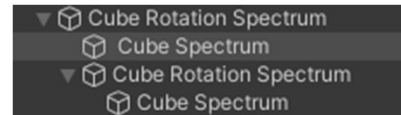
Axe Echelle permet de choisir les axes de changement de taille.

Taille Echelle permet de choisir l'amplitude du changement de taille.

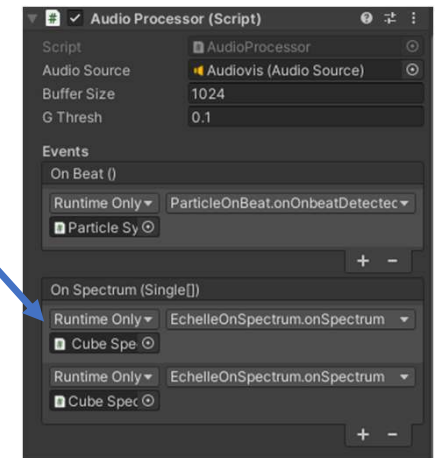
Le script Rotation.cs permet juste de faire tourner un GameObject à la Vitesse Speed et selon l'axe définit.

## GameObject CubeSpectrum

```
0 références  
public void onSpectrum(float[] spectrum)  
{  
    Objet.transform.localScale = new Vector3(TailleEchelle.x * spectrum[0], TailleEchelle.y * spectrum[0], TailleEchelle.z * spectrum[0]);  
    //Objet.transform.localScale = new Vector3(DefaultEchelle.x * spectrum[0], DefaultEchelle.y * spectrum[0], DefaultEchelle.z * spectrum[0]);  
    Debug.Log(DefaultEchelle);  
}
```



## GameObject AudioVis



# GameObject CubeRotation

Ils sont juste là pour tourner et faire joli.