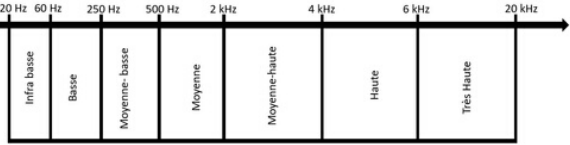
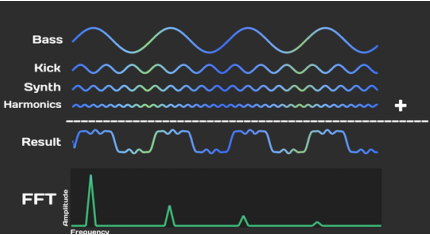


PROJET DE VISUALISATION SONORE, BASÉ SUR L'ANALYSE D'UNE MUSIQUE :

-EXTRACTION DE L'INFORMATION FRÉQUENTIEL PRÉSENTE

-RÉALISATION D'ANIMATION 3D À PARTIR DE UNITY EN ACCORD AVEC L'ANALYSE FRÉQUENTIELLE

Extraction de l'information fréquentiel



GetComponent<AudioSource>()
GetSpectrumData(samples, channel , FFTWindow)

Étape 1
Séparation du spectre fréquentiel en 512 échantillons égaux:
22050 / 512 = 43 Hz par échantillons

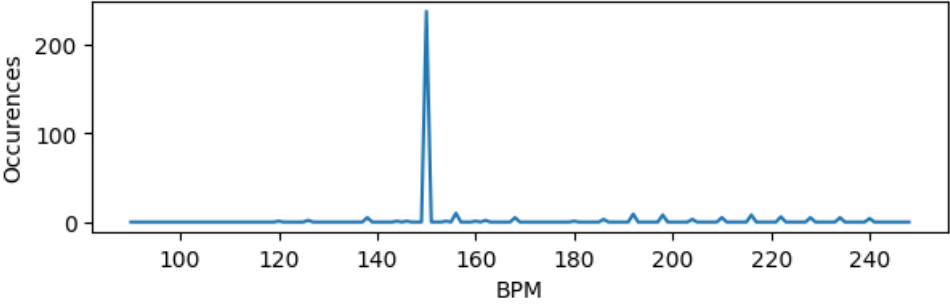
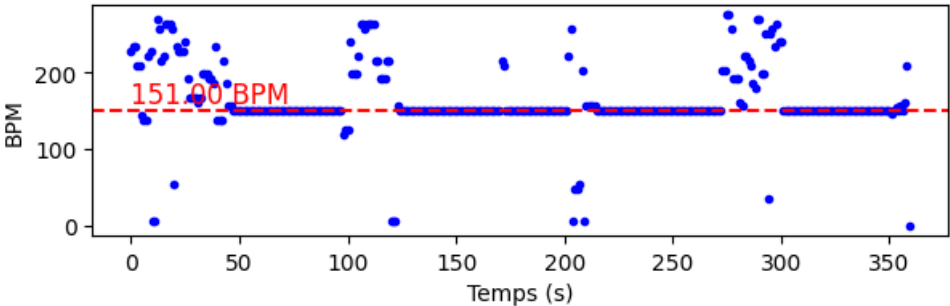
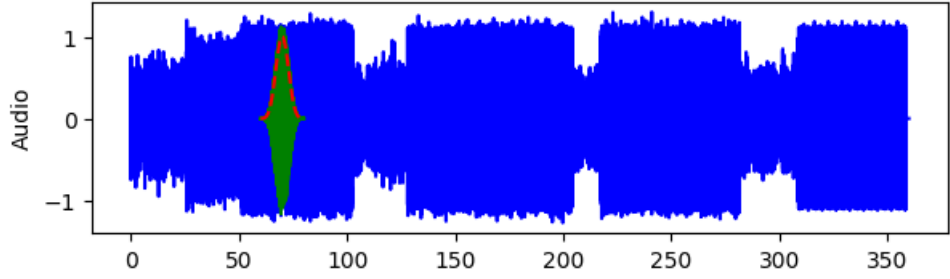
Étape 2
Création de 8 coefficients représentant les 8 plages fréquentielles du spectre de l'audible:
2 éch. = 86 Hz (0)
4 éch. = 172 Hz 87->258 (1)
8 éch. = 344 Hz 259->602 (2)
16 éch. = 688 Hz 603->1290 (3)
32 éch. = 1376 Hz 1291->2666 (4)
64 éch. = 2752 Hz 2667->5418 (5)
128 éch. = 5504 Hz 5419->10922 (6)
256 éch. = 11008 Hz 10923->21930 (7)

Étape 3
Calcul de l'amplitude moyenne permettant de définir le volume globale de la musique

On obtient alors 8 coefficients qui vont pouvoir influencer sur nos animations

Estimation du BPM de la musique

Analyse fréquentielle de la musique



Scène Phyllotaxie

Étape 1

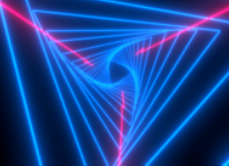

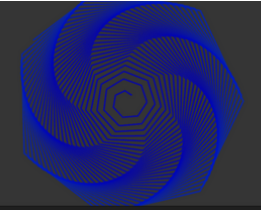
On part d'une équation en coordonnées polaires

angle: $\phi = n * 137.5^\circ$
radius: $r = c \sqrt{n}$

Passage coordonnées polaire à cartésiennes
 $x = r * \cos(\text{angle})$
 $y = r * \sin(\text{angle})$

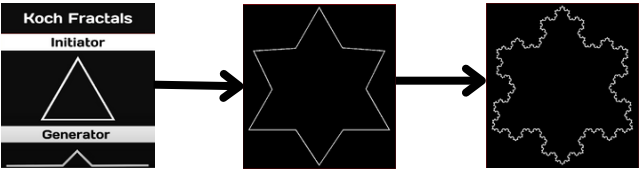
On obtient une fonction avec 3 paramètres en entrée:
le degré, le nombre initial de point et un facteur d'échelle

On l'implémente à l'outil Trail Renderer de Unity qui va à chaque nouvelle frame afficher un nouveau point et tracer un segment entre l'ancien et le nouveau point

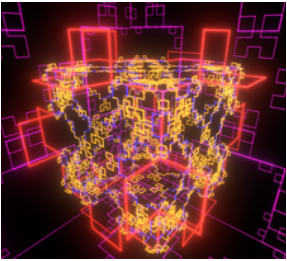
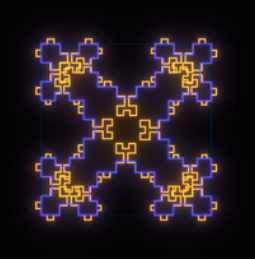



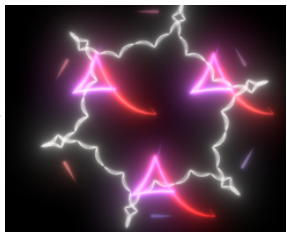

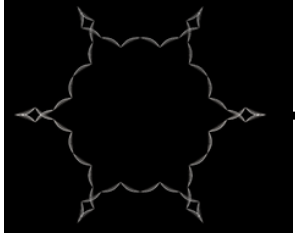
Scène Koch

On a besoin de 2 variables principales, une forme initiale et un générateur.



2 programmes créés, un permettant de générer un fractal, un autre gérant Trail Renderer en utilisant le programme de génération de fractal.





Scène Atomes

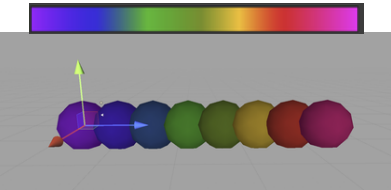
Atomes

Pour chaque atome on spécifie un attracteur et une force d'attraction

Attracteur

Attribution de la couleur sur un gradient en fonction de la bande fréquentielle correspondante

On dispose 8 attracteurs en ligne
Chaque attracteur attire un certain nombre d'atomes



Ajout de plusieurs groupes d'attracteurs et d'une direction de déplacement

Interpolation le long de la ligne de déplacement

Modification taille et couleur des atomes par les coefficients fréquentiels

Perspective d'évolution

Analyse en temps réel d'un flux audio

On ne rentre plus le son dans le logiciel mais le logiciel analysera directement un flux audio provenant d'une entrée micro ou d'une carte son

Utilisation de l'indice de bpm pour gérer la dynamique et l'intensité d'une scène

Diversifier les animations

Amélioration de la communication Python <=> C#

Execution d'un script python depuis un script C# fonctionnelle et récupération du résultat de l'estimation de BPM OK

On doit rentrer manuellement le fichier audio dans le script Python, il serait plus adéquat que le script C# transmette le fichier audio de la classe AudioPeer au script Python sans intervention de l'utilisateur.

