



Édition 2025

PRÉSENTATION DU PROJET

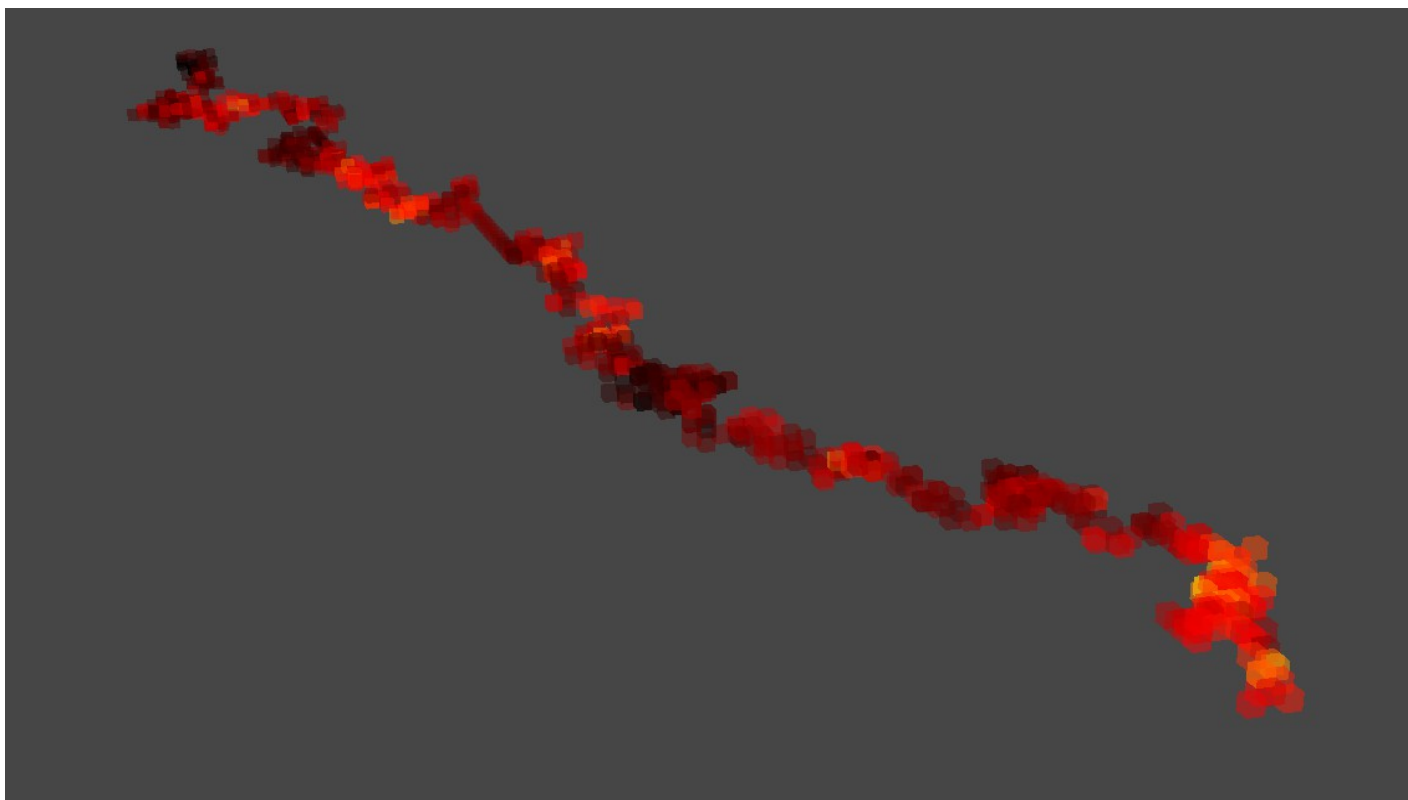


Illustration 1: Vivaldi, les quatre saisons : l'hiver

Nom de votre projet	CubHertz
Membre de l'équipe n°1 (prénom/nom)	Louis Colbert
Membre de l'équipe n°2 (prénom/nom)	Luc Vallet
Niveau d'étude (première ou terminale)	Terminale
Établissement scolaire	Établissement Saint-Alyre, Clermont-Ferrand
Responsable du dépôt (professeur de NSI)	Mr. Laveissière

1 / PRÉSENTATION GÉNÉRALE

Notre projet a pour but de créer une œuvre d'art en trois dimensions qui se construit en temps réel avec le son ambiant. Le résultat se construit donc petit à petit avec une analyse poussée du son. Ce projet est né avec la découverte des œuvres organiques de l'artiste San Base et une question : comment peut-on voir le son ? C'est donc autour de cette question que notre projet s'est orienté. Les objectifs sont donc de réussir à enregistrer le son et de faire des calculs pour afficher une représentation artistique, le tout en temps réel. Ce projet répond donc à une certaine dimension artistique et technique. Nous avons imaginé pouvoir enregistrer des conversations entières, des sons de forêt et d'en avoir une représentation graphique.

2 / ORGANISATION DU TRAVAIL

Notre équipe est composée de deux membres :

- Louis Colbert : Lycéen en terminale, passionné d'informatique et la programmation, j'aimerais devenir ingénieur en informatique à l'ESA. Je me suis occupé de l'analyse du son, et du code permettant la création en trois dimensions.
- Luc Vallet : Dans ce projet, j'ai mis en place et optimisé l'interface graphique ainsi que l'algorithme d'enregistrement du son, avec une grosse partie du travail centrée sur le callback.

Étant dans le même établissement pour les cours de NSI mais dans des villes différentes, nous nous sommes organisés entièrement à distance pour ce projet. Nous avons programmé des réunions avec Google Meet durant nos heures de NSI pour travailler ensemble.

Nous avons passé environ neuf semaines sur le projet.

Pour le partage du code nous avons utilisé Github et créant différentes branches pour éviter des conflits entre nos versions. Nous avons également travaillé avec l'extension LiveShare de Visual Studio Code ce qui nous a permis de coder en même temps sur la même version.

3 / ÉTAPES DU PROJET

Nous avons commencé notre projet par réussir à enregistrer un audio de cinq secondes. Nous l'avons ensuite relié à une interface créée avec le module CustomTkinter permettant de créer des interfaces plus facilement et avec des graphismes plus modernes. L'ajout d'un callback a permis de créer un enregistrement du temps souhaité par l'utilisateur en appuyant simplement sur un bouton. Ensuite différentes idées ont suivi : nous avons d'abord commencé par créer une œuvre en deux dimensions avec un simple fond qui changeait de couleur en fonction de l'intensité de l'audio précédemment enregistré avec le module turtle. Nous avons ensuite cherché à récupérer la fréquence fondamentale et la première fréquence harmonique. Après de nombreuses recherches nous avons décidé d'appliquer la transformée de fourrier pour les trouver. Les fréquences récupérées nous avons commencé à créer des points toujours en deux dimensions pour afficher les fréquences. Seulement l'œuvre n'avait rien d'original, alors nous nous sommes penchés sur un résultat en trois dimensions. Nous avons choisi d'utiliser le module Ursina, célèbre moteur trois dimensions python. Une fois un algorithme déterminé nous l'avons codé en python et le résultat était très satisfaisant. Seulement le résultat n'était toujours pas en temps réel ! Nous avons alors utilisé du multiprocessing avec le module Threading qui permet d'exécuter différentes tâches par le processeur en même temps. Nous l'avons donc utilisé pour enregistrer l'audio tout en faisant les calculs et en affichant le résultat. Nous avons également dû utiliser une queue FIFO (First In First Out) étudié cette année, pour transmettre les données entre les

différentes tâches. Enfin l'algorithme de changement de couleur du fond utilisé en début de projet a été réutilisé et adapté pour ajouter des couleurs à nos cubes. En effet notre organisation avec la création de modules et de classe nous a permis de réutiliser des programmes avec le temps.

4 / FONCTIONNEMENT ET OPÉRATIONNALITÉ

Au moment du dépôt du projet, nous avons fini une première version opérationnelle répondant entièrement à nos attentes. Nous avons décidé de nous attaquer, pour rendre le projet plus autonome, à recoder le module utilisant la transformée de Fourier à l'aide des formules que nous avons étudiées, expliquées par un professeur de mathématiques et de physique.

Pour vérifier l'absence de bugs, nous avons testé notre programme en le soumettant à divers tests. Nous avons enregistré plusieurs centaines d'audios (que nous pouvions soumettre grâce à un bouton test pour choisir un fichier audio) avec des sons très divers et des temps variés. Pour que le programme soit facile d'utilisation, nous avons créé une interface toute simple avec un simple bouton qui permet de lancer l'enregistrement et de l'arrêter.

Nous avons rencontré de très nombreuses difficultés durant le projet : tout d'abord, l'enregistrement faisait planter l'interface. Nous avons alors mis en place un callback, ce qui a réglé le problème. Ensuite, l'analyse du son et l'affichage du fond étaient très lents avec un algorithme qui avait un coût $O(n^2)$ que nous avons réussi à baisser à $O(n)$. Cependant, le programme était toujours trop lent en raison du module turtle qui a des limites graphiques. Le module Ursina, très utilisé pour des jeux vidéo, a éliminé la lenteur du programme, ce qui a permis d'entrevoir le temps réel. Cependant, lors de la découverte du module threading, nous avons passé beaucoup de temps à lire la documentation pour bien l'appliquer à notre projet. Nous avons passé plus de 8 heures à tenter de mettre en place le temps réel, mais Ursina refusait de fonctionner. La découverte d'une fonctionnalité update de Ursina qui se déclenche à chaque frame a permis un grand avancement. La dernière erreur était qu'Ursina faisait appel à une fonction qui ne pouvait fonctionner que dans le thread principal (main). Nous avons donc placé Ursina dans le programme principal et placé l'enregistrement du son dans un nouveau thread (nous avons donc inversé les deux). Et voilà, le programme est fonctionnel. L'ajout des dernières fonctionnalités s'est faites sans bug.

5 / OUVERTURE

Nous avons de nombreuses idées d'ajout de fonctionnalités. Tout d'abord, comme expliqué précédemment, un module de notre soin pour appliquer la transformée de Fourier. Ensuite, nous aimerions ajouter des paramètres extérieurs tels que la luminosité pour la création de l'œuvre. Ensuite, une variation des formes est également une fonctionnalité à moyen terme qui pourrait ajouter une dimension encore plus artistique à notre projet. D'autres fonctionnalités à moyen terme serait l'ajout de paramètres dans l'interface graphique pour augmenter la personnalisation de l'œuvre finale. Enfin, une grosse fonctionnalité qui pourrait être codée à partir de notre projet serait la création d'un monde d'un jeu vidéo à l'aide du son en adaptant l'algorithme : le monde pourrait se créer à mesure que le personnage avance, toujours avec l'audio environnant.

D'un point de vue critique, le projet est un produit fini et fonctionnel, même si des améliorations algorithmiques pourraient être faites, notamment pour éviter l'utilisation de variables globales pour le temps réel (en cours de réflexion). La recherche derrière le projet est poussée, et sa construction en modules permet d'ajouter des fonctionnalités très rapidement.

Si le projet était à refaire, nous aurions commencé directement par utiliser GitHub, qui nous a permis de gagner beaucoup de temps.

Nous considérons avoir gagné de très nombreuses compétences et connaissances : tout d'abord, la lecture de très nombreuses documentations des différents modules choisis nous a donné de bonnes compétences techniques et une compréhension totale des modules. Ainsi que de réussir à trouver les fonctions que nous recherchions et à les adapter au mieux à notre programme. Notre utilisation de classes nous a permis d'améliorer notre maîtrise de celles-ci. Nous avons également découvert le multiprocessing, que nous ne connaissions pas auparavant. Notre gestion des bugs s'est vue grandement améliorée suite à ce projet. Le projet nous a enfin apporté beaucoup de plaisir, nous avons adoré le coder et le voir fonctionner.

Malheureusement, l'absence de filles dans notre groupe de NSI ne nous a pas permis de constituer une équipe mixte.