

## Séquenceur Octobre

Je m'étais fixé comme objectif pour ce mois d'octobre de comprendre le fonctionnement d'un séquenceur qui est un logiciel qui s'avère complexe à prendre en main et de ce fait à coder. J'avais également comme objectifs d'identifier les différentes bibliothèques que j'utiliserai au fur et à mesure du développement de ce projet.

### Fonctionnement du séquenceur

Cette phase d'analyse a permis d'éclaircir les concepts fondamentaux indispensables à la conception d'un séquenceur, conformément aux exigences du cahier des charges. La distinction entre une Piste (ligne temporelle pour organiser des sons) et un Pattern (motif musical répétitif) a été établie. J'ai également étudié les mécanismes de Bouclage (Loop) d'un fragment musical et de la gestion du Tempo (BPM), éléments cruciaux pour la lecture en temps réel et la synchronisation des pistes. Ces concepts sont désormais clairs pour pouvoir commencer le codage.

### Bibliothèques

J'ai choisi d'utiliser le framework PySide pour ce projet, celui-ci m'offrira une assez grande liberté pour réaliser tous mes objectifs et est notamment plus poussé que Tkinter et plus adapté que PyGame qui sont deux autres bibliothèques auquel je pensais.

Pour l'audio, j'utiliserai la bibliothèque sounddevice, qui est plus moderne et facile à prendre en main que pyaudio. Quant au chargement des différents sons, la bibliothèque soundfile me permettra d'utiliser les différents fichiers audios. Enfin, j'utiliserai la bibliothèque json pour le chargement et la sauvegarde des différents projets. Finalement, d'autres bibliothèques python standards seront probablement utilisées et documentées dans le développement du projet.

### Suppléments

J'ai également mis en place un dépôt GitHub pour ce projet. Même si je réalise ce projet seul, il me permettra de travailler de manière efficace sur mes différentes machines et également de prendre en main git qui est un outil très utile dans le monde du développement informatique et qui m'est et sera utile au fil des années.

Ce dépôt GitHub contient pour le moment l'arborescence complète du projet, qui sert de base à la structure modulaire définie. Ce dépôt, bien que personnel, est géré selon un workflow strict pour garantir un historique clair et maintenir la branche principale stable et fonctionnelle.

De plus, une convention de codage a été mise en place afin de garantir la lisibilité et la maintenabilité du code.

J'ai également utilisé « venv » pour installer les différentes bibliothèques externes que je vais utiliser lors du développement de ce projet. Même si cela n'est pas très utile dans mon cas car je n'ai pas un

grand nombre de projet python sur ma machine, il est considéré bonne pratique d'installer les bibliothèques externes dans des environnements virtuels comme venv pour éviter les « dependency hell » (si j'utilise par exemple pour ce projet une certaine version de PySide et sur un autre une version différente, étant donné que je ne peux avoir qu'une seule version de cette bibliothèque installée sur ma machine en local, l'utilisation des venv prend tout son sens.

Enfin, pour préparer la phase d'implémentation de décembre, les assets audio nécessaires ont été acquis. Plusieurs collections de sons de musique électronique (notamment des one-shots de kicks, snares, hi-hats, basses et synthés) ont été téléchargées et organisées dans un dossier assets/ pour être immédiatement utilisables par le moteur audio.