

## Séquenceur janvier

Conformément à la conclusion du rapport de décembre, le mois de janvier était dédié à la résolution de la limitation technique majeure du prototype : la gestion de la polyphonie. L'objectif principal était de développer une synchronisation audio parfaite entre les pistes pour éviter les coupures de son lors des chevauchements. En parallèle, le planning prévoyait l'amélioration de l'interface graphique via l'ajout d'un retour visuel dynamique (curseur de lecture) pour faciliter le repérage dans la grille rythmique.

### Refonte du moteur audio

Pour dépasser les limitations de la lecture séquentielle simple observées en décembre (coupure du son précédent lors d'un nouveau déclenchement), j'ai procédé à une refonte complète du cœur audio (MoteurAudio).

- Architecture Callback : J'ai abandonné la lecture bloquante pour une architecture asynchrone basée sur une Callback. Le moteur audio n'attend plus d'ordre de lecture direct mais alimente un flux continu (Stream) géré par la carte son.
- Mixage Mathématique (NumPy) : Conformément aux prévisions, j'ai intégré la bibliothèque scientifique NumPy. Elle permet d'effectuer une addition mathématique des signaux audio en temps réel. Concrètement, si un Kick et un Snare tombent sur le même temps, leurs valeurs d'ondes sont additionnées numériquement avant d'être envoyées au flux de sortie
- Résultat : Cette approche garantit désormais une polyphonie illimitée et une stabilité parfaite, même à des tempos très élevés (200 BPM), sans aucune coupure sonore.

### Évolution de l'Interface et Expérience Utilisateur

L'interface graphique a évolué pour transformer le prototype en un instrument jouable et réactif :

- Feedback Visuel: J'ai implémenté un système de suivi visuel qui illumine la colonne active de la grille en temps réel. Cela permet à l'utilisateur de savoir exactement quel temps est en train d'être joué, comblant le manque de repères visuels de la version de décembre.
- Contrôle du Tempo : L'intervalle de lecture n'est plus fixe. J'ai ajouté un contrôleur dynamique (Slider) permettant de faire varier le tempo de 60 à 200 BPM. Une formule mathématique ( $\$Intervalle = 60000 / BPM / 4\$$ ) recalcule instantanément la vitesse du QTimer sans interrompre la lecture.

## Qualité du Code et Refactoring

Profitant de la fin de cette phase de développement critique, j'ai effectué un nettoyage global du code (Refactoring). Les commentaires superflus ont été remplacés par des docstrings techniques standardisées, et la gestion des dépendances a été mise à jour (ajout de numpy dans requirements.txt) pour garantir la portabilité du projet.

## Conclusion

Les objectifs de janvier sont pleinement atteints. Le logiciel a franchi un cap technique décisif : il n'est plus un simple lanceur de sons séquentiel, mais un véritable moteur de mixage temps réel stable.

Avec un moteur audio robuste et une interface fonctionnelle, le projet est prêt pour la prochaine étape prévue au cahier des charges : la persistance des données. Le mois de février sera donc consacré à l'implémentation de la sauvegarde et du chargement des projets (format JSON), afin de permettre à l'utilisateur de conserver ses compositions.