BENSAFIA
L3 SDN
CHAHINE

Cahier des charges – Séquenceur de musique électronique en Python

Introduction

Ce projet consiste à créer un séquenceur de musique électronique en Python. L'utilisateur pourra composer des rythmes et des mélodies, organiser plusieurs pistes et les lire en boucle. Le rendu sonore doit se faire en temps réel. L'interface sera graphique dès la première version pour simplifier la manipulation des patterns et des pistes.

Présentation du projet

Le séquenceur permettra de créer, éditer et lire des patterns rythmiques et mélodiques. L'utilisateur pourra sauvegarder et charger ses projets. Le logiciel doit rester modulable pour intégrer facilement de nouvelles fonctionnalités et extensions.

Objectifs principaux

- Créer et modifier plusieurs pistes (kick, snare, hihat, basse, synthétiseur). -
- Organiser des patterns pour chaque piste et les lire en boucle.
- Synchroniser les pistes pour un rendu audio fluide.
- Sauvegarder et charger les projets.
- Garantir une faible latence pour que le son réagisse immédiatement.
- Maintenir une architecture modulaire pour faciliter l'ajout de nouvelles fonctions.

Fonctionnalités secondaires et extensions possible

- Effets audio simples : reverb, delay, filtres.
- Export multipiste en fichier audio (WAV).
- Annuler ou refaire des actions.
- Indicateurs visuels pour la latence et la charge CPU.

Interface et expérience utilisateur

-Zoom et défilement dans la timeline **:** pour naviguer facilement dans les patterns longs.
-Drag & drop : déplacer des patterns ou des samples avec la souris.
-Color coding : couleurs différentes pour les pistes, patterns ou types de sons.
- Feedback visuel sur le tempo : un indicateur de battement ou un « playhead » qui se déplace.
-Contrôles simples : lecture, pause, volume et tempo accessibles directement.

BENSAFIA
L3 SDN
CHAHINE

Contraintes techniques

- Langage : Python 3.x
- Compatibilité Windows et Linux.
- Bibliothèques audio possibles : pyaudio, sounddevice, pygame.mixer.
- Interface graphique : PyQt ou PySide (Tkinter ou pygame pour prototype).
- Lecture en temps réel avec synchronisation précise.
- Sauvegarde des projets au format simple (JSON).

Description du logiciel

Fonctionnalités principals

- Gestion de pistes et patterns.
- Lecture en boucle avec tempo ajustable.
- Interface simple pour contrôler lecture, volume et tempo.
- Sauvegarde et chargement des projets.

Fonctionnalités secondaires (extensions)

- Ajout d'effets audio.
- Export multipiste en fichier audio (WAV).
- Interface graphique avancée avec boutons, sliders et retour visuel.

Exemple d'usage

- Créer un pattern rythmique.
- Définir le tempo et la durée de la boucle.
- Jouer la boucle en continu.
- Ajouter des pistes mélodiques ou de basse.

Développement technique

Structure du code

- **Core** : gestion des patterns, des pistes, du tempo et du séquenceur principal.
- **AudioEngine** : lecture des samples et synchronisation.
- **Interface** : entrée utilisateur (/GUI), affichage et contrôle, (affichage console pour les tests)

BENSAFIA
L3 SDN
CHAHINE
Tests

- Tests unitaires sur la logique des patterns, pistes et tempo.
- Tests fonctionnels sur la lecture en temps réel.
- Tests utilisateurs sur l'ergonomie et la fluidité.

Planning prévisionnel

En octobre, je me concentrerai sur l'analyse et l'exploration. Cette phase consistera à comprendre le fonctionnement d'un séquenceur (pistes, patterns, tempo, bouclage) et à identifier les bibliothèques audio et graphiques pertinentes pour le développement.

En novembre, je mettrai en place un premier prototype minimal. Il permettra la lecture d'un sample audio et disposera d'une interface graphique simple avec les commandes de base.

Le mois de décembre sera consacré à l'implémentation de plusieurs pistes et à l'introduction des patterns. J'ajouterai également une première gestion de la lecture en boucle.

En janvier, je travaillerai sur la synchronisation entre les pistes et sur l'évolution de l'interface graphique, avec un affichage en grille pour organiser visuellement les samples.

Au cours de février, je commencerai à développer les fonctionnalités avancées, notamment la lecture en boucle stable et le système de sauvegarde et de chargement des projets.

En mars, je continuerai sur la lancée de février et je finaliserai les choix concernant l'implémentation de toutes les fonctionnalités avancées.

Le mois d'avril sera dédié à la documentation technique et à la rédaction du rapport, afin de présenter clairement le fonctionnement et les choix réalisés. Egalement, je me concentrerai sur la stabilisation et finalisation du logiciel : correction des bugs et amélioration de l'interface.

Enfin, en mai, je préparerai la soutenance en finalisant la présentation et en effectuant les derniers ajustements du logiciel.

Livrables

- Code source complet.
- Documentation technique avec diagrammes.

BENSAFIA
L3 SDN
CHAHINE

- Manuel utilisateur.
- Rapport de tests.

Glossaire

- Séquenceur : logiciel pour programmer et jouer des suites de sons ou de notes.
- **Pattern** : motif musical ou rythmique répétitif. Une piste peut contenir plusieurs patterns.
- **Piste** : ligne temporelle pour organiser des sons ou des patterns. Ce n'est pas un son.
- **Sample** : enregistrement audio élémentaire (kick, clap, note). Pas un extrait d'une autre musique.
- **Tempo (BPM)** : vitesse de la musique en battements par minute.
- **Boucle (Loop)** : fragment musical répété en continu.
- **Effet audio** : transformation d'un son pour changer son rendu (reverb, delay, filtre).
- **Latence** : délai entre une action et le son produit.
- **Rendu en temps réel** : son produit immédiatement, avec un délai de calcul quasi nul (ms).
- **Synchronisation** : alignement des pistes et patterns pour qu'ils jouent ensemble correctement.
- Playhead : indicateur qui montre la position actuelle de lecture sur la timeline.