**Sujet : Automate pour la Génération de Musique**

Ce mini-projet consiste à utiliser un automate pour générer des séquences musicales en fonction de règles définies, explorant ainsi la musique algorithmique. L'automate représente les différents motifs musicaux et les transitions entre eux, basés sur des règles harmoniques et rythmiques. Les événements déclencheurs peuvent inclure des signaux externes ou des déclencheurs internes, déterminant les changements de motifs et les variations musicales.

Ce mini-projet offre aux étudiants l'occasion de combiner la créativité musicale avec les concepts d'automates finis, explorant ainsi les possibilités de la génération de musique assistée par ordinateur.

**Les Objectifs du projet**

Générer des séquences musicales automatiquement

Utiliser les automates finis pour représenter et gérer les motifs musicaux

Intégrer des règles harmoniques et rythmiques pour guider la génération musicale

Permettre des déclenchements basés sur des signaux externes ou internes.

**2. Recherche et Planification**

**Étude Préliminaire**

* **Musique algorithmique** : Comprendre les bases de la musique générée par des algorithmes.
* **Automates finis**: Etudier comment les automates peuvent représenter des transitions entre différents états.

**Planification**

* **Diagramme d’état** : Dessiner un diagramme d’état représentant les différents motifs musicaux et les transitions entre eux.
* **Règles et Transition** : Définir les règles harmoniques et rythmiques qui guideront les transitions d’un motif à un autre

**3. Conception de l’automate**

Modélisation des Motifs Musicaux

* Chaque état de l’automate correspond à un motif musical spécifique
* Les transitions entre les états représentent les changements de motifs musicaux.

**Définition des Evènements déclencheurs**

**Signaux externes :** Par exemple, des entrées de capteurs, des interactions utilisateur.

**Déclencheurs internes :** Par exemple, la fin d’un motif, des conditions temporelles.

**4. Développement**

**Choix des outils et Technologies**

**Langage de programmation** : Python

**Bibliothèques musicales** : MIDIUtil (Python)

**Implémentation**

* **Automate** : Programmer l’automate avec ses états et transitions.
* **Génération musicale** : Ecrire des fonctions pour générer des motifs musicaux.
* **Gestion des évènements** : Implémenter la logique pour gérer les signaux externes et internes.

**Phase initiale**

**Automate** : Programmer l’automate avec ses états et transitions.

Notre automate est caractérisé par :

* Les **Etats** 1(do), 2(ré), 3(mi), 4(fa), 5(sol), 6(la), 7(si)
* **L’Alphabet** (a(do), b(ré), c(mi), d(fa), e(sol), f(la), g(si))
* Les **transitions**

(1, a, 1) (1, b, 2) (1, c, 3) (1, d, 4) (1, e, 5) (1, f, 6) (1, g, 7)

(2, a, 1) (2, b, 2) (2, c, 3) (2, d, 4) (2, e, 5) (2, f, 6) (2, g, 7)

(3, a, 1) (3, b, 2) (3, c, 3) (3, d, 4) (3, e, 5) (3, f, 6) (3, g, 7)

(4, a, 1) (4, b, 2) (4, c, 3) (4, d, 4) (4, e, 5) (4, f, 6) (4, g, 7)

(5, a, 1) (5, b, 2) (5, c, 3) (5, d, 4) (5, e, 5) (5, f, 6) (5, g, 7)

(6, a, 1) (6, b, 2) (6, c, 3) (6, d, 4) (6, e, 5) (6, f, 6) (6, g, 7)

(7, a, 1) (7, b, 2) (7, c, 3) (7, d, 4) (7, e, 5) (7, f, 6) (7, g, 7)

* Les **Etats Initiaux**
* Les Etats terminaux
  1. Développons les fonctionnalités principales suivantes :
     + Rendre un automate déterministe s’il ne l’est pas

**Vérifions d’abord si l’automate est déterministe**

Pour qu’un automate fini soit déterministe :

* Chacun de ses états doit avoir au plus une transition pour chaque symbole de l'alphabet.
* Il ne doit y avoir qu'un seul état initial.

Codons d’abord une fonction qui permet de vérifier si un automate est déterministe. Pour cela, nous allons utiliser la définition précédente.

L’algorithme est le suivant :

**Détermination de l’automate non déterministe**

Pour déterminiser un automate, nous allons utiliser l’algorithme de déterminisation.

**Étapes de la déterminisation**

1. **Initialiser l'état initial du nouvel automate** avec l'ensemble des états initiaux de l'automate original.
2. **Pour chaque état du nouvel automate (qui est un ensemble d'états de l'automate original)**, ajouter des transitions pour chaque symbole de l'alphabet, en utilisant les transitions de tous les états de l'ensemble.
3. **Si un nouvel ensemble d'états est généré** par une transition, ajouter cet ensemble comme un nouvel état dans le nouvel automate.
4. Répéter le processus jusqu'à ce qu'aucun nouvel ensemble d'états ne soit généré.

Minimisation d’un automate

Pour minimiser un automate, il faut tout d’abord : - S’assurer qu’il est déterministe. - Éliminer les états qui ne sont pas atteignables à partir de l’état initial, ainsi que les transitions adjacentes à ces états. - Éliminer les états à partir desquels on ne peut pas atteindre un état final, ainsi que les transitions adjacentes à ces états.

