LES TROPHÉES NSI

Édition 2024

DOSSIER DE CANDIDATURE

PRÉSENTATION DU PROJET

nom de votre projet :	Music Matrix
membres de l'équipe :	François Deffayet
membres de l'équipe :	Brieuc Guillet de Chatelus
membres de l'équipe :	Augustin Dunglas
Niveau d'étude :	première
établissement scolaire :	Saint Jean Hulst
enseignant de NSI :	Mr de Regis

> PRÉSENTATION GÉNÉRALE :

Lorsque notre professeur nous a parlé des trophées NSI, nous avons longtemps cherché une idée de projet qui serait à la fois relativement technique, à notre niveau et en accord avec les critères des projets pouvant être présentés. Nous avons tout d'abord pensé à développer notre propre réseau social dédié au partage de fiches de cours dans le cadre de notre lycée et, grâce à un chat, avec la possibilité de parler à d'autres élèves afin de pouvoir obtenir de l'aide pour les cours. Cependant, ce premier projet a été terminé relativement rapidement et nous nous sommes rendus compte qu'il ne correspondait pas tout à fait au critère demandé : en effet, nous nous basions sur un serveur FTP réalisé à l'aide d'une Raspberry Pi 5, il y avait donc de gros problèmes de sécurité.

Voyant que cette première idée n'était pas forcément la meilleure, nous nous sommes mis à réfléchir à une autre idée de projet et c'est ainsi qu'Augustin, un des membres de notre groupe, a eu l'idée de faire une mini IA capable de composer de la musique avec Python. Étant tous les trois musiciens (violoniste, guitariste et violoniste, pianiste), cette idée de faire s'associer musique et informatique nous a tout de suite plu. Ainsi, après quelques recherches sur le domaine des IA en musique, nous sommes tombés sur le nom de Markov, un mathématicien russe qui a découvert un modèle de suite de probabilité : Les Chaines de Markov permettant de décrire une composition d'une partition grâce à des lois de probabilités. La première partie de notre projet a donc consisté à coder une « IA » de Markov capable de composer une petite musique à partir d'une note de départ et d'une longueur de partition donnée : cette partie du code a été réalisée par François.

Cependant, cette première étape du projet nous sembla encore une fois trop simple, c'est pourquoi, pour complexifier la chose tout en restant dans le thème de la musique, nous avons décidé mutuellement, sous l'impulsion de Brieuc, le troisième membre du groupe, d'élargir le projet en codant toujours avec Python une petite application permettant de composer de la musique en s'inspirant, pour l'interface, d'un doodle fait par google en « date ? ». Ainsi notre projet final est-il une application permettant de composer sa propre musique en s'inspirant, si on le souhaite, de petites mélodies générées par l'IA de Markov, puis d'imprimer les partitions et de jouer la musique composée. Si on le souhaite, on peut aussi enregistrer l'audio générée lors de l'écoute de notre musique : cette application aura pour nom Music Matrix.

> ORGANISATION DU TRAVAIL :

Notre équipe est composée de trois personnes :
Augustin Dunglas : passionné d'informatique et de programmation depuis la 6ème, j'ai commencé par prendre des cours sur des plateformes gratuites en ligne comme Openclassrooms sur divers langages de programmation (HTML, CSS, Java et Python mais je me suis plutôt concentré sur Python). A partir de la 3ème, j'ai découvert le monde de la cybersécurité grâce à des sites de « capture the flag » qui permettent d'apprendre la cybersécurité tout en s'amusant comme Rootme ou Cyberini. J'aime beaucoup ce domaine et j'aimerais en faire mon métier. Dans le projet pour le trophée NSI, j'ai eu l'idée de faire une application de musique et je me suis occupé du développement de toute l'interface utilisateur permettant de composer la musique et d'interagir avec Music Matrix.
Brieuc Guillet de Chatellus : j'aime beaucoup l'informatique et je suis passionné d'aéronautique (mention bien au BIA). J'aimerais devenir ingénieur dans ce domaine. Je me suis occupé de coder toute la partie du logiciel permettant de jouer la musique que l'utilisateur a composée et de régler les différents paramètres de lecture de la musique.
François Deffayet : passionné d'informatique et d'électronique depuis la troisième, je suis à Saint-Jean depuis la seconde et je rêve de faire une MP2I ou une école d'ingénieurs pour ensuite travailler dans la cybersécurité ou bien être ingénieur en robotique. Dans ce projet, je me suis chargé de coder l'IA de Markov et ensuite de toute la partie du code permettant d'écrire la partition et ensuite de l'enregistrer.

Au total, nous avons passé sur ce projet chacun de notre côté plus de 70 heures pour arriver à un résultat satisfaisant. Lors du développement de cette application, nous avons principalement codé sur Thonny Python, sur Visual studio code et sur Anaconda. Nous nous sommes aussi beaucoup renseignés sur la musique en midi sur des forums comme Github et sur les travaux de Markov sur Wikipédia.

LES ÉTAPES DU PROJET :

Le développement de Music Matrix s'est déroulé en trois étapes principalement.

En effet, nous avons d'abord eu l'idée de ne faire qu'une IA de composition de musique avec Python mais, voyant que cela était trop dur et long à faire, nous nous sommes tournés vers un modèle plus simple d'IA, à s'avoir les chaînes de Markov, permettant de composer une musique grâce à l'étude de probabilités de suites de notes et de rythmes. Cette partie du code fonctionne sur une succession de tests

afin de déterminer quelle note a le plus de chance de venir après une autre, et de même pour les rythmes.

Après avoir développé cette petite IA, nous avons décidé de complexifier la chose en ajoutant la possibilité pour l'utilisateur de composer sa propre musique. Pour cette partie du projet nous nous sommes inspirés de l'interface d'un doodle. L'interface utilisateur de cette partie du projet fonctionne grâce à la librairie Tkinter permettant de créer des interfaces interactives et relativement simples à comprendre ; la deuxième partie de l'interface, à savoir la création de la partition une fois la musique composée, repose sur la librairie Turtle. Nous avons fait le choix d'utiliser cette librairie car elle permet de donner à l'utilisateur l'impression que la partition est écrite par une main invisible qui dessine sous ses yeux la musique qu'il vient de composer. En outre, elle nous permet de rendre notre code plus complet en utilisant un large panel de bibliothèques ainsi que leurs fonctionnalités respectives. Cependant, un des inconvénients de cette librairie est que l'impression de la partition est un peu lente, mais d'une façon générale, elle permet d'obtenir une partition avec un effet de « fait main » assez sympathique.

Enfin, la dernière partie du code, qui concerne la lecture du morceau composé, est gérée par deux bibliothèques Mido et Midiutil : la première nous permet de jouer les morceaux générés par l'IA de Markov et de jouer les différents timbres lors du choix de la sonorité de l'instrument, et la deuxième (Midiutil) permet de jouer le morceau final. En effet, c'est une bibliothèque permettant de combiner plusieurs pistes Midi en un seul fichier audio et ensuite de le jouer ou bien de l'enregistrer. Ainsi, la réalisation du code s'est déroulée en trois étapes : l'IA de Markov, l'interface graphique de composition et enfin la lecture de la partition composée.

> FONCTIONNEMENT ET OPÉRATIONNALITÉ :

La réalisation de ce projet a été un véritable parcours du combattant, surtout vers la fin de sa réalisation : nous avons en effet chacun développé notre partie du code de notre côté puis nous les avons rassemblées, le véritable enjeu a alors été de faire cohabiter les différentes parties de code en attribuant à chaque variable des noms spécifiques et clairs, en mettant certaines parties du code dans des fonctions qui interagissent les unes avec les autres en fonction des actions de l'utilisateur. Ainsi la partie d'assemblage du projet aura sûrement été la plus complexe et la fonction « print() » nous aura été bien utile pour résoudre les bugs rencontrés au fur et à mesure de l'assemblage. Aujourd'hui, le code tourne parfaitement et sans erreur.

Un autre enjeu de ce projet était de faire une application facile à utiliser et donc en accord avec les conformités demandées. Il nous semble que nous nous sommes bien débrouillés et que, si l'interface n'est pas parfaite, elle est tout de même assez facile et compréhensible grâce à des noms clairs et précis.

Enfin, la principale difficulté rencontrée aura été de jouer la partition. Quelques jours avant la fin du projet, nous avons découvert un bug majeur dans la lecture du morceau composé et sa résolution nous a occupés tous les trois pendant

les derniers jours avant la remise des projets. Il existe en effet très peu de documentation sur les librairies permettant de jouer plusieurs piste Midi à la fois. Cela nous a beaucoup compliqué la tâche car nous avons dû lire le code de la librairie Midiutil pour essayer d'en comprendre les différentes fonctionnalités et syntaxe.

> OUVERTURE :

Lors de la fin de la réalisation du projet, nous avons rencontré de nombreuses difficultés pour jouer la musique. La bibliothèque Midiutil que nous utilisons n'est probablement pas idéale mais, pressés par le temps, nous avons fait le choix de la garder. Cependant, à l'avenir, nous essayerons d'utiliser un autre système plus simple plus rapide pour générer la musique : la technologie Midi n'est pas des plus récentes et elle est assez limitée dans de nombreux domaines comme le nombre d'instruments disponibles.

Dans le futur, les modifications que nous comptons réaliser sont l'amélioration de la création du fichier audio et l'ajout d'une interface plus jolie et plus interactive. Une autre possibilité d'amélioration serait de créer une IA plus développée et véritablement capable de composer une musique avec plusieurs instruments et un plus grand panel de notes et de figures rythmiques. On pourrait utiliser, par exemple, une IA fonctionnant sur un réseau de neurones génératifs afin qu'elle puisse apprendre de ses erreurs et composer de façon toujours plus intelligente. Une autre idée d'amélioration majeure serait de permettre à l'utilisateur de garder en mémoire un projet pour ensuite le continuer plus tard. Ainsi, ce projet d'application musicale est loin d'être terminé et de nombreuses idées d'amélioration nous viennent en tête. Par manque de temps, nous n'avons malheureusement pas pu les mettre dans ce projet mais nous espérons qu'à l'avenir nous pourrons apporter ces modifications. La réalisation de ce projet nous a permis de découvrir, en effectuant de nombreuses recherches, l'aspect musical de Midi et des IA de génération de musique et de création de fichiers audio, d'autant plus que cette partie de Python permettant de faire de la musique Midi est très peu documentée sur internet. La réalisation du projet a donc été relativement difficile mais les difficultés l'ont aussi rendue très stimulante.

D'un point de vu de l'utilité du projet nous avons choisit de développer cette application en voyant qu'il n'existe pas ou peu de logiciel gratuit sur internet pour composer et écouter de la musique. Ainsi cette application permet de faire dans les grandes lignes ce que permettent de faire des applications comme Musecore mais qui sont-elles payantes, Music Matrix permet donc à des musiciens débutants et sans budget de composer et d'écouter leur premier morceau.

> RESUMÉ DU PROJET

Pour notre projet du concours NSI 2024, nous avons développé, à l'aide du langage de programmation Python, une application que nous avons appelée Music Matrix. Cette application a pour fonction principale de permettre à l'utilisateur de composer de la musique pour 15 instruments au maximum et de pouvoir ensuite écouter et imprimer la partition qu'il vient de composer. Dans cette première partie du code, nous utilisons pour jouer la musique composée la bibliothèque Midiutil, qui permet de jouer des mélodie Midi combinant plusieurs instruments. Pour l'impression de la partition, nous avons choisi d'utiliser la bibliothèque Turtle, qui donne une impression de « fait main » assez plaisante. Enfin, pour l'interface utilisateur, nous utilisons la librairie Tkinter qui est simple et extrêmement complète à notre niveau. Cependant, Music Matrix propose aussi à l'utilisateur la possibilité d'écouter une mélodie ayant une longueur maximale de 50 notes créée par une petite IA générative fonctionnant grâce aux suites des Markov qui permettent de calculer des probabilités de suites mélodiques et rythmiques. Pour le fonctionnement de cette partie du code, nous utilisons la librairie Random permettant de simuler des situations de probabilité et donc d'utiliser les chaînes de Markov. Pour jouer la mélodie générée, nous utilisons la bibliothèque Mido permettant de jouer des sons Midi avec un seul instrument.

Ainsi l'application Music Matrix permet de composer, jouer et imprimer des partitions mais aussi de générer de petites mélodies simples pour, par exemple, donner de l'inspiration au compositeur en manque d'idées. Pour finir, nous envisageons à l'avenir deux améliorations principales pour ce projet : tout d'abord, permettre à l'utilisateur d'enregistrer un projet en cours pour ensuite le poursuivre plus tard, et, ensuite, améliorer la petite IA générative afin que l'utilisateur puisse donner un avis sur les mélodies générées afin que l'IA apprenne de ses erreurs et tienne compte des remarques faites par l'utilisateur. Une autre idée d'amélioration serait de rendre l'interface plus ludique et simple à utiliser qu'elle ne l'est actuellement.