

INSA Lyon - Département Informatique

Rapport

de

PROJET DE FIN D'ÉTUDES

**Développement d'interface graphique technologie web -
faisabilité et évaluation**

***Development of web-based graphical user interface -
feasibility and evaluation***

Colin THOMAS

Soutenance le 26 juin 2020

Projet réalisé du **5 février 2024** au **26 juillet 2024**

dans la structure d'accueil

Arturia (Grenoble)

Référent	: Yann GRIPAY, Maître de Conférences	INSA Lyon
Tuteur	: Timothée BÉHÉTY, Référent Technique Logiciels Compagnon	Arturia (Grenoble)

Contents

1	Introduction.	1
1.1	Mise en contexte.	1
1.2	Définition du problème.	1
1.3	Aperçu des contributions.	1
1.4	Problématique et plan.	2
2	Contexte.	2
2.1	Contexte du stage.	2
2.1.1	Intégration à l'équipe Compagnon	2
2.1.2	Méthodologie	2
2.2	Arturia.	2
2.3	Présentation du sujet de stage.	3
2.3.1	Mission.	3
2.3.2	Intérêt.	4
3	Etude bibliographique et analyse de l'existant.	4
3.1	L'Arturia Software Center existant.	4
3.2	Le Midi Control Center existant.	5
3.3	L'API Web MIDI.	6
3.4	Définitions nécessaires.	6
4	Propositions scientifiques et techniques.	7
4.1	Méthodologie de travail.	7
4.1.1	Organisation de l'équipe et réunions.	7
4.1.2	Git et revues de code.	8
5	description gemaps	8
	References bibliographiques	9

Annexes

A	Données collectées.	11
B	Éléments majeurs de la conception.	11

1 Introduction.

1.1 Mise en contexte.

J'ai réalisé mon stage de fin d'études chez Arturia, une société Grenobloise, fabricant d'instruments de musique numériques et électroniques, du 5 février au 26 juillet 2024. C'était l'entreprise que je souhaitais intégrer pour ce stage de fin d'études, non seulement car j'utilise et apprécie les produits et logiciels Arturia, mais aussi car je souhaitais me rapprocher du milieu de ma passion pour la musique. J'ai intégré pendant ce stage l'équipe Compagnon d'Arturia, qui m'a proposé un sujet de prototypage d'interfaces Web pour la nouvelle génération de deux de leurs logiciels, l'Arturia Software Center et le Midi Control Center. J'ai pu ainsi participer à la conception d'un logiciel modulaire rassemblant ces deux applications, développer des interfaces C++ préexistantes en Web, concevoir de nouvelles interfaces et prototyper l'utilisation d'APIs Web novatrices. Ce stage me permet de mettre en oeuvre les compétences de développement Web acquises pendant mon stage précédent, ainsi que de mettre en pratique les connaissances théoriques que j'ai pu apprendre pendant mes études.

1.2 Définition du problème.

L'Arturia Software Center (ASC) et le Midi Control Center (MCC) sont deux logiciels vieillissants, qu'Arturia souhaite mettre à jour. Pour faire cela, l'entreprise souhaite mettre à l'épreuve de nouvelles technologies de développement graphique.

En effet, ces logiciels ont été développés respectivement en 2014 pour l'Arturia Software Center, et en 2013 pour le Midi Control Center. A ce moment, l'équipe Compagnon que j'intègre pour ce stage n'existait pas encore, et ces logiciels ont été développés dans des technologies qui ne sont potentiellement pas optimales pour ces programmes.

Ainsi, l'équipe Compagnon a décidé de prototyper une application modulaire, nommée GeMAPS, qui rassemblerait l'Arturia Software Center et le Midi Control Center. Ce logiciel distribuerait et présenterait des pages Web pour ses interfaces utilisateur.

Le défi de mon équipe est donc de réussir à adapter un logiciel préexistant dans une nouvelle technologie, en en séparant le front-end et le back-end, et en réussissant à faire une application suffisamment modulaire pour qu'elle puisse intégrer à l'avenir de nouveaux éléments non encore définis.

1.3 Aperçu des contributions.

Lors de ce stage, j'ai pu travailler sur plusieurs éléments :

- Démontrer l'utilisation de multiples Frameworks Web (React, Angular, Vue, Vue Nuxt) dans une application modulaire C++ JUCE.
- Développer toutes les interfaces existantes C++ de l'Arturia Software Center en Vue Nuxt.

- Concevoir et développer les interfaces un Midi Control Center entièrement en Web, qui permet de configurer le clavier maître Arturia MiniLab3.
- Concevoir et développer l'interface du logiciel modulaire accueillant le Midi Control Center et l'Arturia Software Center.
- Prototyper l'implémentation de l'API Web MIDI pour communiquer avec les appareil Arturia en front-end Web uniquement.
- Expérimenter l'utilisation de l'API Web USB pour mettre à jour en Device Firmware Update les appareils Arturia, ainsi que de l'API FileSystem.

La majeure partie de mon travail était concentrée sur le développement des interfaces Web du Midi Control Center, ainsi que de l'Arturia Software Center, en étroite collaboration avec le membre de l'équipe travaillant sur le back-end.

1.4 Problématique et plan.

La problématique de ce rapport de stage concerne d'une part, l'intégration d'interfaces Web dans un logiciel modulaire, et d'autre part, l'objectif de déplacement de l'intelligence logicielle du Midi Control Center dans le Web, impliquant l'expérimentation et la démonstration d'utilisation de l'API de diverses technologies Web novatrices.

2 Contexte.

2.1 Contexte du stage.

2.1.1 Intégration à l'équipe Compagnon

Pour ce stage, j'ai été intégré à l'équipe Compagnon d'Arturia, une sous-équipe de l'équipe Système au sein du département R&D. Sous la supervision de mon tuteur, Timothée Béhéty, qui dirige cette sous-équipe composée de 4 membres, j'ai collaboré étroitement avec Jean-Yves Tissot. Il est responsable du développement du back-end C++ de l'Arturia Software Center (ASC) et est le développeur du projet GeMAPS. Étant donné que l'équipe Compagnon se concentre exclusivement sur le développement en C++. J'ai sollicité l'expertise de Fabian Boilay, membre de l'équipe du site web d'Arturia, pour les revues de code.

2.1.2 Méthodologie

2.2 Arturia.

Arturia est un fabricant d'instruments de musique numériques et électroniques. La société est basée à Grenoble et développe des reproductions logicielles d'anciens synthétiseurs (analogiques

et numériques), des claviers MIDI, des contrôleurs, des synthétiseurs analogiques et des interfaces audio.

Il s'agit d'une entreprise employant 180 personnes, qui a été fondée en 1999. A sa création, ses deux fondateurs Frédéric Brun (actuel CEO d'Arturia) et Gilles Pommereuil ont commencé par créer des reproductions de synthétiseurs analogiques réputés au format virtuel, en commençant par développer l'Analog V, inspiré par les synthétiseurs Moog et en collaboration avec leur créateur. De nombreuses autres reproductions ont suivi, et sont devenues par la suite la V Collection, qui est la collection d'instruments virtuels d'Arturia, et la FX Collection, qui est son équivalent pour les effets audio.

Arturia a depuis élargi son catalogue à des instruments virtuels qui lui sont propres, plusieurs synthétiseurs analogiques renommés, différents contrôleurs MIDI ainsi que des cartes sons.

L'équipe Compagnon dans laquelle je réalise mon stage s'occupe des logiciels accompagnants les produits Arturia, comme le gestionnaire de licences, le logiciel de configuration de matériel, ou encore des logiciels spécifique à chaque produits, comme par exemple l'application mobile qui accompagne le clavier AstroLab.

2.3 Présentation du sujet de stage.

2.3.1 Mission.

L'Arturia Software Center (ASC) et le Midi Control Center (MCC) sont deux logiciels vieillissants, qu'Arturia souhaite mettre à jour. Pour faire cela, l'entreprise souhaite mettre à l'épreuve de nouvelles technologies de développement graphique.

En effet, ces logiciels ont été développés respectivement en 2014 pour l'Arturia Software Center, et en 2013 pour le Midi Control Center. A ce moment, l'équipe Compagnon que j'intègre pour ce stage n'existait pas encore, et ces logiciels ont donc été développés par l'équipe Software, qui se spécialise dans le développement d'instruments virtuels. Ils ont opté pour l'utilisation de technologies de développement d'instruments virtuels, notamment le framework C++ JUCE, pour la création de ces outils. Cependant, une mise à jour de l'ASC et du MCC en utilisant des technologies différentes permettrait de se détacher des outils spécifiquement conçus pour les logiciels audio, potentiellement inadaptés à ces applications.

Ainsi, l'équipe Compagnon a décidé de prototyper une application modulaire qui rassemblerait l'Arturia Software Center et le Midi Control Center. Cette application serait capable de lancer des bibliothèques C++ en tant que back-end, de servir en HTTP des pages Web statiques en tant que front-end, et d'afficher un host permettant de naviguer sur ces pages Web. De cette manière, l'Arturia Software Center et le Midi Control Center constitueraient = deux modules de cette application, chacun dotés d'un back-end sous forme de bibliothèque C++, et d'un front end sous forme de page web statique, communiquant avec leur back-end par requêtes HTTP.

2.3.2 Intérêt.

Utiliser des interfaces web pour ces logiciels se révèle ainsi positif sur plusieurs points :

- Les technologies web sont plus efficaces pour créer rapidement des interfaces. En effet, les bibliothèques et outils pré-développés et libres d'accès sont nombreux, en constante évolution, et facile d'utilisation : on peut prendre pour exemple la fonctionnalité multilangue, qui est une fonctionnalité qui prend de l'importance pour Arturia, et qui demande beaucoup de ressources, pour un développement plus facile, classique et très documenté en technologies Web. De même, les framework Web offrent souvent une courbe d'apprentissage plus faible que le C++ ou autre langage qui n'est pas haut-niveau.
- Ceci permet, pour développer ces interfaces, de pouvoir rechercher des profils de développeurs front-end Web, ce qui est plus facile que de devoir rechercher des développeurs C++, puis de les former sur le framework JUCE, pour ensuite travailler sur des interfaces. En séparant le développement front-end du reste de l'application, ceci permet de rechercher des développeurs spécialisés dans leur domaines, et ainsi une meilleure évolutivité de l'équipe.
- Ensuite, cela permet de séparer le front-end du back-end. En effet, avoir une interface HTTP entre un back-end et un front-end permet de bien séparer ces deux parties. Ceci amène de la modularité et de la scalabilité, car chaque partie peut être modifiée sans affecter l'autre. On peut imaginer également une plus grande flexibilité technologique, car l'application Web pourrait être développée dans n'importe quel framework, sans impliquer de changement pour les librairies C++. Enfin, en isolant les accès directs au back-end, on peut ainsi mettre en place de meilleures mesures de sécurité pour protéger les logiciels et les licences utilisateurs.

L'Arturia Software Center étant l'outil qui permet de gérer l'installation des logiciels, un back-end est forcément nécessaire pour soutenir le front-end. En effet, par design et pour des raisons de sécurités, les technologies Web n'ont pas accès au système de fichiers de l'utilisateur.

En revanche, la question se posera pour le Midi Control Center, dont la principale fonctionnalité est de communiquer par protocole Midi avec le contrôleur. Nous étudierons pendant ce stage la faisabilité d'un Midi Control Center entièrement réalisé en technologies Web.

3 Etude bibliographique et analyse de l'existant.

3.1 L'Arturia Software Center existant.

L'Arturia Software Center ¹ (ASC) est le logiciel de gestion de licences et de téléchargement de software d'Arturia : il est nécessaire pour tout utilisateur souhaitant utiliser un logiciel Arturia. L'utilisateur peut naviguer parmi les logiciels disponibles sur l'Arturia Software Center sur la

¹<https://www.arturia.com/support/asc-arturiasoftwarecenter>

page "Explore Products", activer la licence du logiciel sur la page "My Products", le télécharger, le mettre à jour, gérer l'emplacement de ses fichiers dans les paramètres, ainsi que de rentrer le code d'un produit Arturia physique afin de bénéficier des logiciels qui l'accompagnent.

L'Arturia Software Center est composé de deux parties :

Tout d'abord, une partie principale, qui est ce logiciel qui permet de gérer ses licences et ses téléchargements de logiciels Arturia, et fournit une interface graphique.

Ensuite, un logiciel nommé ASC Agent, qui fonctionne en arrière plan, et vérifie les licences utilisateur à chaque démarrage de produit Arturia, afin d'éviter toute fraude et téléchargement illégal.

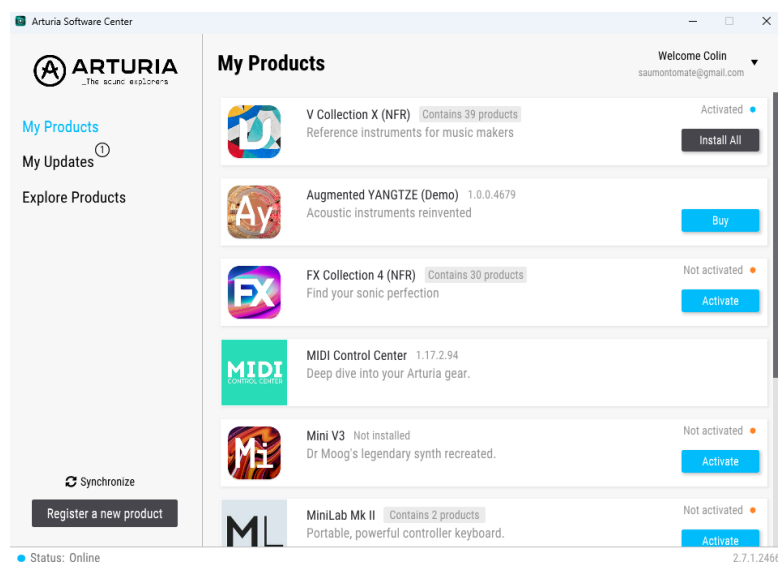


Figure 1: Arturia Software Center (ASC) existant

3.2 Le Midi Control Center existant.

Le Midi Control Center ² (MCC) est le logiciel de configuration de matériel Arturia : il permet de gérer les configurations et les paramètres des synthétiseurs et contrôleurs MIDI, ainsi que de faire les mises à jour firmware pour les synthétiseurs analogiques, cartes sons et claviers midis physiques d'Arturia. Il communique en norme MIDI avec les produits Arturia pour leur configuration.

Il est composé d'une page "Device Settings" permettant de modifier les paramètres généraux du contrôleur actuellement connecté (par exemple, son temps de mise en veille ou ses courbes de vélocité), une page "Controller Map" permettant de modifier les paramètres de chaque contrôle du contrôleur, (par exemple, modifier la note attribuée à une touche en particulier), et une page permettant d'importer ou d'exporter des templates de paramètres. Il est également

²<https://support.arturia.com/hc/fr-fr/sections/4405740766098-MIDI-Control-Center>

possible d'effectuer une détection automatique de mises à jour disponibles pour le firmware du contrôleur.



Figure 2: Midi Control Center (MCC) existant

3.3 L'API Web MIDI.

L'API Web MIDI ³ fournit une manière pour les applications web d'interagir avec les périphériques MIDI connectés à l'ordinateur. Elle permet d'envoyer et de recevoir des messages MIDI, par exemple recevoir des messages de Note On / Note Off, mais aussi envoyer des messages Sysex permettant de modifier les paramètres du périphérique.

Actuellement, cette API n'est disponible que sur les navigateurs Safari et Chromium. Cependant, cela ne pose aucun problème pour notre application, car nous avons la liberté de choisir le navigateur. En l'occurrence, celui que nous utilisons est celui de Juce, et il est basé sur Chromium pour Windows, et sur Safari pour Mac.

3.4 Définitions nécessaires.

Nous allons ici définir quelques termes propres au matériel de production musicale.

Synthétiseur analogique ⁴ : Un synthétiseur analogique est un synthétiseur qui utilise des circuits analogiques et des signaux électriques analogiques pour générer des sons par des techniques électroniques. Contrairement à un synthétiseur numérique, qui stocke et communique l'information en binaire, l'information sonore est transmise dans l'analogique en tant que variation de tension.

³https://developer.mozilla.org/en-US/docs/Web/API/Web_MIDI_API

⁴https://fr.wikipedia.org/wiki/Synthétiseur_analogique

MIDI ⁵ : Le Musical Instrument Digital Interface ou MIDI est un protocole de communication et un format de fichier dédiés à la musique, et utilisés pour la communication entre instruments électroniques, contrôleurs, séquenceurs, et logiciels de musique. Il s'agit du protocole standard du matériel électronique de musique. Il permet par exemple à un Contrôleur MIDI d'envoyer le signal qu'une certaine note a été pressée avec une certaine vélocité, permettant à un logiciel sur ordinateur de produire un son en conséquence. Dans le sens inverse, ce protocole permet à l'utilisateur de paramétrer son contrôleur MIDI en envoyant des signaux SYSEX, comportant les informations nécessaires sur le paramètre à modifier et la valeur à lui attribuer.

Contrôleur MIDI ⁶ : Un contrôleur MIDI est un appareil utilisé pour envoyer des signaux MIDI à l'ordinateur où à un autre appareil (par exemple, un synthétiseur analogique). Il peut se présenter sous la forme d'un clavier de piano, de faders, de potentiomètres, d'encodeurs, ou bien d'autres.

Synthétiseur virtuel : Un synthétiseur analogique virtuel est un synthétiseur simulant le son de synthétiseurs analogiques par le biais de processeurs numériques. Il peut être utilisé sous la forme de Stand-Alone, c'est à dire sans logiciel qui l'entoure, en le contrôlant par exemple par un contrôleur MIDI, mais également et plus populairement sous la forme Plug-In, c'est à dire en tant que module à intégrer à un logiciel de Musique Assistée par Ordinateur.

Message Sysex ⁷ : Un message SysEx est un type de message MIDI permettant de régler les paramètres d'un périphérique MIDI, ce qui n'est pas adapté à la syntaxe normale du protocole MIDI.

4 Propositions scientifiques et techniques.

4.1 Méthodologie de travail.

4.1.1 Organisation de l'équipe et réunions.

L'équipe Compagnon dans laquelle j'ai travaillé se réunissait pour un stand-up chaque matin, où l'on disait ce qu'on a fait la veille, ce qu'on prévoyait de faire le jour-même. C'était aussi l'occasion de partager les problèmes rencontrés et de recueillir les avis de l'équipe.

Nous effectuions également une réunion d'architecture du projet GeMAPS tous les vendredis, où l'on concevait ensemble le projet sur lequel moi et Jean-Yves travaillions.

⁵https://fr.wikipedia.org/wiki/Musical_Instrument_Digital_Interface

⁶<https://troidecis.fr/musique/instrument-midi-definition>

⁷https://steinberg.help/cubase_artist/v10.5/fr/cubase_nuendo/topics/midi_editors/midi_editors_sysex_messages_c.html

4.1.2 Git et revues de code.

Pour mon projet, j'ai fonctionné avec Git. Travaillant seul, j'avais une branche principale, et une branche secondaire sur laquelle je faisais des modifications avant de faire une Merge Request (MR). Ainsi, je pouvais, pour des modifications importantes, demander la revue de Fabian Biolay sur mes modification avant de les intégrer à ma branche principale. J'ai eu ainsi 3 projets Git, que sont la page hôte qui accueille les deux applications, l'Arturia Software Center, et le Midi Control Center.

5 description gemaps

Unified Control Center : L'Unified Control Center est l'application hôte qui affiche les pages web des modules. Nous utilisons pour l'instant le navigateur intégré au framework C++ Juce, qui intègre Internet Explorer (lui-même basé sur Chromium) sous Windows et WebKit sous Mac. Nous envisageons, pour le futur, d'utiliser Chromium Embedded Software (CEF), afin d'unifier les navigateurs sous tous les systèmes d'exploitation, et également afin d'avoir un contrôle plus grand sur le navigateur.

GeMAPS : Generic Modular Architecture with Pluggable Services (GeMAPS) est l'architecture du programme qui lance les librairies C++ des modules, et distribue leurs pages Web.

Module : Un module est un dossier, composé d'un fichier de configuration décrivant le module, un dossier "lib" où l'on peut trouver les librairies C++ qui seront lancées par Gemaps (.dll dans le cas de Windows, et .dylib sur Mac), et un dossier "web" où l'on peut trouver les pages web statiques qui seront distribuées par Gemaps. Un module peut avoir aucune, une ou plusieurs librairies C++, et elles seront toutes lancées au lancement de Gemaps. Un module peut également avoir aucune une ou plusieurs page Web, dont les URL à servir sont décrites dans le fichier de configuration du module.

Listing 1: Programme inconnu

```
import numpy as np

def incmatrix(genl1, genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m, 1), int) #dummy variable

    #compute the bitwise xor matrix
    M1 = bitxormatrix(genl1)
```

```

M2 = np.triu(bitxormatrix(genl2),1)

for i in range(m-1):
    for j in range(i+1, m):
        [r,c] = np.where(M2 == M1[i,j])
        for k in range(len(r)):
            VT[(i)*n + r[k]] = 1;
            VT[(i)*n + c[k]] = 1;
            VT[(j)*n + r[k]] = 1;
            VT[(j)*n + c[k]] = 1;

        if M is None:
            M = np.copy(VT)
        else:
            M = np.concatenate((M, VT), 1)

VT = np.zeros((n*m,1), int)

return M

```

References

testtesttest

Annexes

Annexe A: Données collectées.

Annexe B: Éléments majeurs de la conception.

Annexe C: Preuves de théorèmes.

Développement d'interface graphique technologie web - faisabilité et évaluation

Colin THOMAS

Résumé

Durant mon stage de fin d'études, j'ai pu participer à la conception et au développement d'interfaces Web pour la nouvelle génération des Logiciels Compagnon d'Arturia. Ces logiciels sont l'Arturia Software Center, le magasin de logiciels d'Arturia, et le Midi Control Center, le logiciel de configuration de matériel Arturia. Dans l'optique de mettre à jour des logiciels vieillissant, et d'expérimenter de nouvelles technologies de développement graphique, mon équipe a décidé de prototyper une application modulaire les rassemblant, et d'utiliser des interfaces Web pour la partie front-end.

Mon rôle a été de développer des interfaces Web pour l'Arturia Software Center, en y intégrant un back-end C++ développé par un autre membre de l'équipe. J'ai également réalisé un nouveau design pour le Midi Control Center, et ai pu prototyper l'utilisation de l'API Web MIDI pour communiquer avec les appareils Arturia.

J'ai pu par ailleurs, au long de mon stage, suivre et participer à la conception de l'application modulaire lors de réunions d'architecture hebdomadaires.

Travailler sur un projet aussi innovant à Arturia m'a permis d'expérimenter des domaines novateurs, mais aussi d'approfondir mes compétences de travail en équipe et de communication que j'ai pu développer lors de mes précédents stages.

Mots-clés : Web ; Vue Nuxt ; Cross-Plateforme ; Web APIs ; Application Modulaire.

Abstract

During my final internship, I had the opportunity to participate in the design and development of web interfaces for Arturia's next generation Companion Softwares. These softwares include the Arturia Software Center, Arturia's software store, and the Midi Control Center, Arturia's hardware configuration software. With the aim of updating aging software and experimenting with new graphic development technologies, my team decided to prototype a modular application bringing them together, utilizing web interfaces for the front-end.

My role involved developing web interfaces for the Arturia Software Center, integrating a C++ back-end developed by another team member. I also redesigned the Midi Control Center and prototyped the use of the Web MIDI API to communicate with Arturia devices on the web, as well as experimenting with the Web USB API to update device firmware.

Throughout my internship, I followed and participated in the conception of the modular application during weekly architecture meetings.

Working on such an innovative project at Arturia allowed me to explore innovative domains and deepen my teamwork and communication skills, which I had developed during my previous internships.

Keywords: Web ; Vue Nuxt ; Cross-Platform ; Web APIs ; Modular App.