

Rapport Final Visuals Music

Goodwin Stéphane

Gisselmann Arthur

Perrin Océane

2019/2020 – Info 2



Affiche 1: Affiche réalisée en communication

Table des matières

Introduction.....	3
Contexte.....	3
Résumé de l'existant.....	3
Conception.....	4
Répartition des tâches.....	4
Partie rédactionnelle.....	4
Partie Informatique.....	4
Méthodes.....	5
Sprint “1”.....	5
Sprint “2”.....	6
Sprint “3”.....	7
Résultats.....	11
Conclusion.....	12
Bilan du travail / Perspectives.....	12
Bilan personnel.....	13
Annexe.....	14

Introduction

Contexte

Il s'agit de programmer un outil qui permet de visualiser une chanson. Le programme prend en entrée un morceau de musique, l'analyse en temps réel (amplitude, fréquence, etc.) et génère une sortie visuelle à l'aide d'infographies 2D et 3D abstraites. Le développement peut être fait progressivement en commençant par produire un simple audiogramme, graphe en deux dimensions montrant l'amplitude du son en fonction du temps. Ensuite, un spectrogramme, ou sonagramme, qui permet de visualiser l'intensité du son par fréquence en fonction du temps. L'analyse pourra être affinée à volonté. Par exemples, les sons harmoniques peuvent être associés à des formes lisses et les sons inharmoniques à des formes rugueuses. La forme, la couleur et la texture des objets tridimensionnels, ainsi que leur position dans l'espace, sont autant de variables que l'on peut associer pour représenter le timbre, par exemple.

L'objectif est de lire une musique que l'utilisateur apprécie et de visualiser cette dernière sur un affichage de bonne qualité.

Résumé de l'existant

On trouve sur le marché de plus en plus de logiciels qui permettent de visualiser la musique. Aussi, il est très courant de voir des personnes, comme des DJs, chercher de nouveaux moyens afin d'améliorer leurs performances et de gagner en qualité de présentation visuelle. Les clients possèdent des attentes sur un logiciel permettant de visualiser de la musique. Effectivement, lorsqu'une personne la visualise, elle souhaite l'entendre avec une cohérence entre le son et l'image affichée.

L'année 2018 est un record pour le marché des applications sur mobiles, ce record augmente d'année en année. 50% des téléchargements mobiles se situent uniquement en Chine. Les recettes de 2018 vont jusqu'à 101 milliards de dollars pour l'App Store, Google Play Store et les boutiques Android tiers. En 2016, on avait que 40 milliards de dollars. C'est une énorme augmentation. Pour 2019, les analyses montrent une année toute aussi bonne, on atteindrait les 120 milliards de dollars.

20% des personnes qui utilisent leurs smartphones, l'utilisent pour écouter de la musique. C'est 9 millions de titres de musique qui sont téléchargés sur mobile et 16,4% de chiffre d'affaire du marché de la musique numérique qui est réalisé seulement par les téléphones mobiles.

Il y a beaucoup voir énormément de personne dans le monde qui écoute de la musique. On estime une écoute d'environ deux heures par jour par personne. Certaines personnes ne peuvent pas se passer de musique lors d'une journée, cela permet aux utilisateurs de se détendre, de se concentrer, ou de penser à autre chose ...

Nous souhaitons visés des personnes qui se situe dans la tranche d'âge des 16 à 24 ans, car ces personnes écoutent le plus souvent de la musique sur leur téléphone. Ils sont plus susceptible d'utiliser notre application, même si les autres personnes d'autres tranches d'âges peuvent l'utiliser mais il serait moins nombreux.

Nous avons donc une augmentation forte sur l'utilisation des téléphones, notre projet consistera à développer une application sur ordinateur, puis par la suite de la développer sur Android.

Conception

Le cahier des charge, les diagrammes UML ainsi que le planning sont disponible en [Annexe](#).

Répartition des tâches

Partie rédactionnelle

Tableau 1: Représente la répartition entre les membres du groupe du travail rédactionnel

	Études Préalables				Diagrammes UML		Affiche	Anglais		Fin du projet	
	Études du Besoin	Études de l'existant	Cahier des charges	Planning	Classe	Cas d'utilisation		Rapport	Slideshow	Rapport	Slideshow
Océane	X					X		X		X	X
Arthur		X					X	X			
Stéphane		X	X	X	X			X	X	X	X

Partie Informatique

Tableau 2: Représente la répartition entre les membres du groupe du travail informatique

	Sprint 1	Sprint 2	Sprint 3
Océane		X	
Arthur			
Stéphane	X	X	X

Méthodes

Pour ce projet, nous devons appliquer les méthodes agiles. Le projet était donc réparti sur trois sprints d'environ 2 semaines séparés d'une semaine pour présenter notre avancement à notre tuteur.

Sprint "1"

Pendant ce sprint, nous avons commencés à implémenter, l'interface graphique avec une ligne au milieu comme nous pouvons le voir ci-dessous. Après avoir créer l'interface graphique, nous avons fait en sorte de pouvoir lire un fichier disponible sur la machine de l'utilisateur. A partir de là, on ne pouvait lire que des fichiers avec comme extension ".wav"

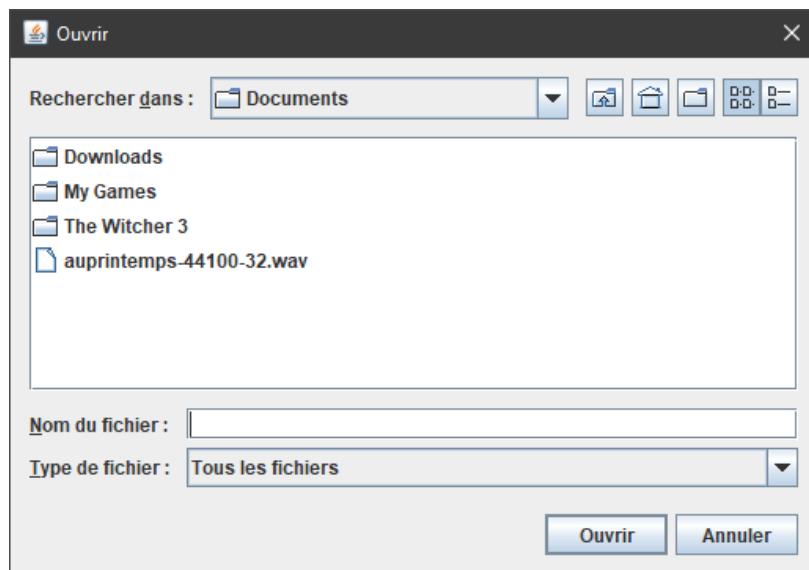


Illustration 1: FileFinder permettant de sélectionner un fichier au choix

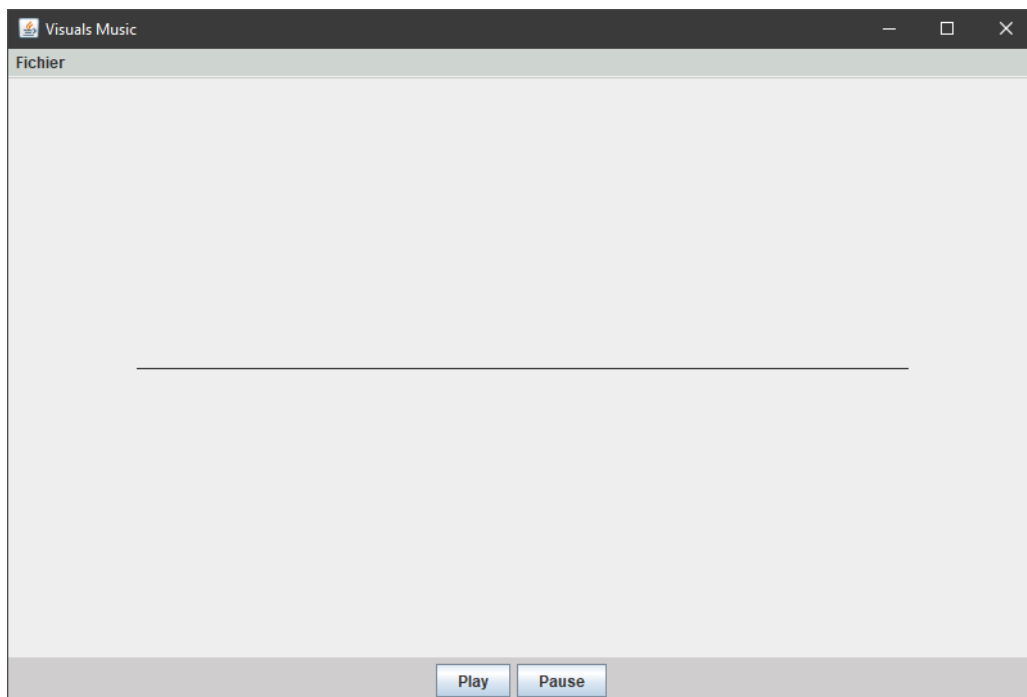


Illustration 2: L'interface graphique pendant le sprint 1

Sprint “2”

Pendant ce sprint, on a réunis les Bouton “Play” et “Pause”. Quand on clique sur ce bouton, le texte change. Aussi, on a inséré un bouton “Stop”, qui permet d'arrêter l'écoute et de revenir au début de la chanson et un bouton “Ecran” qui est sensé mettre la fenêtre en plein écran, cependant, le bouton ne fonctionne pas. Par la suite, on s'est intéressé à la visualisation de la musique. Cependant, comme à la suite de la lecture d'un fichier audio, on obtient des octets et non la fréquence, on a du réaliser une transformé de Fourier rapide (FFT). Afin de réaliser la FFT, nous avons utilisé une librairie Java, “flanagan”. Pour faire simple, une FFT est une suite de formules mathématiques complexes qui prend des octets et les transforme en nombres complexes. Dans notre cas, les nombres complexes obtenus ont leurs parties imaginaires nulles, donc si on la tronque, on obtient la fréquence. A partir de ce point, nous pouvions réellement implémenter les visualiseur 2D et 3D. C'est pourquoi nous avons ajouté un menu dans la barre du haut. Celui-ci permet de naviguer entre l'affichage 2D et 3D. Le visualiseur 3D ne fus implémenter qu'au sprint “3”.

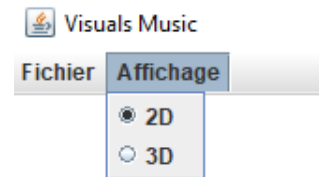


Illustration 3: Accès au changement de visualiseur

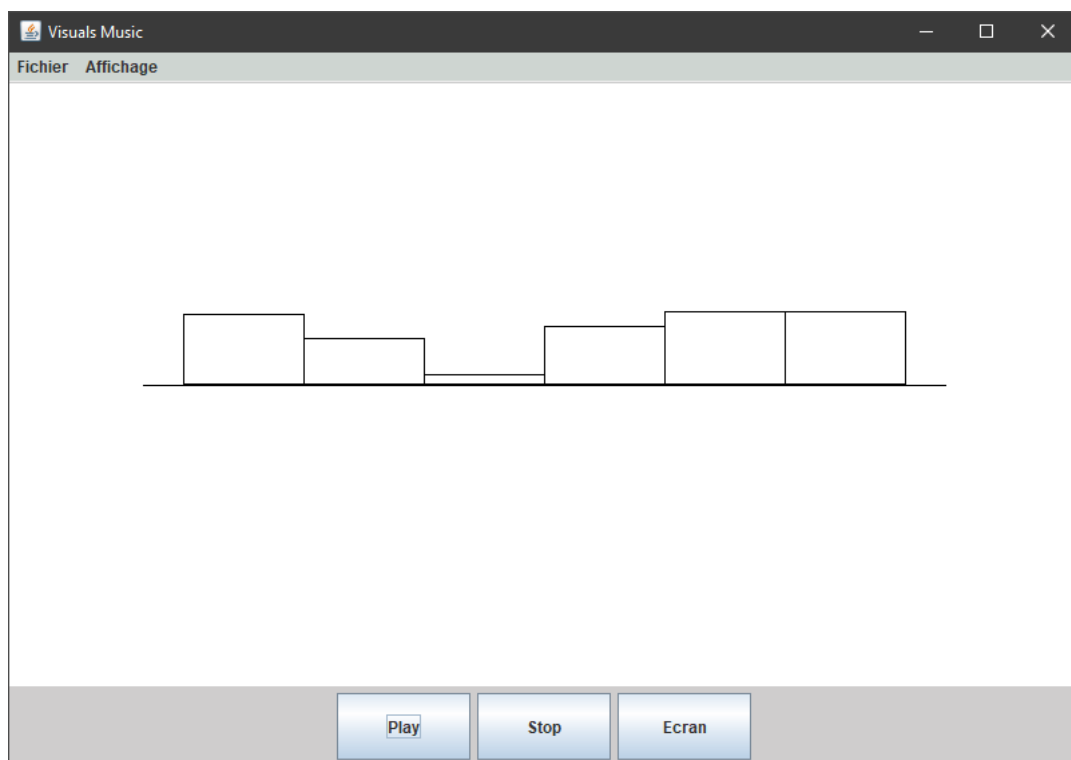


Illustration 4: L'interface graphique pendant le sprint 2

De plus, dans ce Sprint nous avons créé un backlog, composé de six colonnes. Les trois premières permettent de voir l'avancement du projet. Composées de checkboxes, nous étions capable de signaler rapidement l'état de la tâche au reste du groupe. Aussi, en fonction de la case cochée, la ligne entière changeait de couleur. Les deux suivantes permettaient de signaler la priorité ainsi que la difficulté de la tâche sur une échelle allant de zéro à cinq. Pour finir la dernière colonne représentait la tâche à faire.

Tableau 3: Quelques éléments du backlog

Etat		Priorité	Difficulté		
Fait	En Cours	A Faire	(0-5)	(0-5)	TODO
		X	4	5	Générencense de la musique (tout type de format/fréquence/encodage)
		X	3	2	Background paramètres
	X	X	5	4	Rapport final
	X	X	5	2	Correction fautes Rapports
X	X	X	5	5	Diapo Anglais 06/01/20
X	X	X	5	5	Affichage 3D / librairie JOGL / opengl
X	X	X	5	5	FFT
X	X	X	5	4	uml
X	X	X	5	4	Affichage 2D
X	X	X	5	4	Affiche communication

Sprint “3”

Le sprint “2” finis, nous pouvions nous pencher sur le visualiseur 3D. Nous avons effectué des recherches afin de savoir comment faire de la 3D. Nous avons donc été redirigé vers les librairies JOGL, Java3D et Canva3D. Ne sachant pas laquelle utiliser, nous avons demandé à notre tuteur afin de savoir laquelle de ces librairies étaient la plus pertinente. Cette dernière nous a conseillé JOGL. Nous avons donc installé les librairies relatives à JOGL et nous avons pu implémenter le visualiseur 3D. La tâche fut complexe au début, le temps que le fonctionnement du traçage 3D soit compris.

Ensuite, nous voulions être capable de modifier l’affichage, notamment l’affichage 2D. C’est pourquoi, nous avons implémenter des boutons afin de pouvoir modifier l’amplitude et la taille des rectangles ainsi que l’espace entre ceux-ci.

Aussi, notre tutrice, trouvant notre interface graphique triste, nous a demandé de rajouter des couleurs, ce que nous avons fait. Nous étions partis pour donner une couleur aléatoire à chaque nouveau rectangle. Après quelques tests, nous nous sommes dit qu'il serait préférable que l'utilisateur puisse choisir entre des couleurs aléatoires ou des couleurs souhaitées.

A partir de là, il aurait été compliqué et non ergonomique d'ajouter directement la possibilité de modifier les couleurs dans la fenêtre principale. C'est alors que nous avons créé une nouvelle fenêtre "Paramètres" accessible via un menu dans la barre du haut.

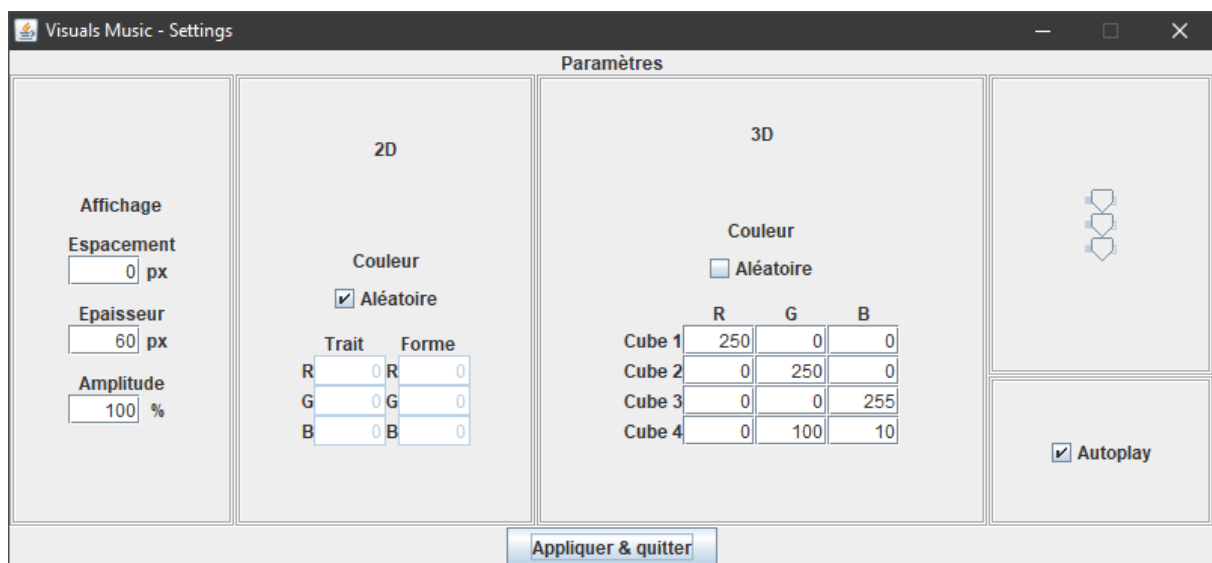


Illustration 5: Fenêtre des paramètres

Depuis cette fenêtre, l'utilisateur peut modifier les couleurs des dessins 2D et 3D, l'espacement, l'épaisseur et l'amplitude ainsi que signaler s'il souhaite ou non que la musique soit en "autoplay" - lecture automatique de la musique dès que le fichier audio est ouvert. Afin de faciliter la disposition de tous les éléments, nous avons créé un fichier Excel représentant l'interface graphique avec un GridBagLayout.

Tableau 4: Principe du GridBagLayout sur la fenêtre des paramètres

Paramètres											
Affichage			2D			3D			Aigu		
Espacement			Couleur			Couleur			Grave		
		px	Aléatoire			Aléatoire			Vitesse		
Epaisseur			Trait		Forme				R	G	B
		px	R		R		Cube1				
Amplitude			G		G		Cube2				
		%	B		B		Cube3				
							Cube4				
Autoplay											
Appliquer et quitter											

Nous avons également réalisé un slider afin de pouvoir modifier le volume de la musique. Celui-ci n'est pas réellement lié au volume mais plutôt au gain sonore.

Aussi nous avons modifié l'affectation de la fréquence qui se base sur la moyenne de toutes les fréquence d'un son, maintenant elle se base sur la fréquence maximale d'un son. Ceci permet d'admirer un net changement de hauteur d'un rectangle à l'autre. En effet l'affichage était très linéaire avant ce changement.

Visualiser la musique (2019 – 2020)

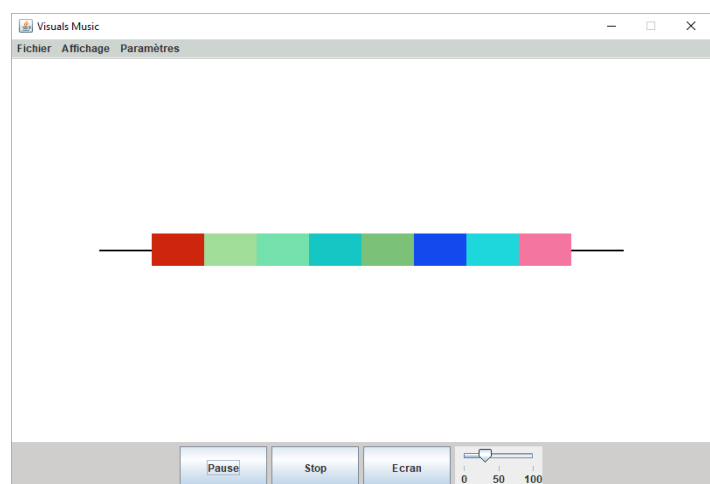


Illustration 6: L'interface graphique 2D pendant le sprint 2 qui était très linéaire 26

Par la suite, nous avons créé des raccourcis claviers pour notre application. Dorénavant, dès que l'utilisateur appuie sur Espace, la musique se pause ou se dé-pause, et dès qu'il presse Echap, l'application se ferme.

Dans une de nos première version du visualiseur 3D, les cubes se coupaient. Nous avons modifié cela afin de bien les séparer et d'apprécier d'autant plus l'affichage.

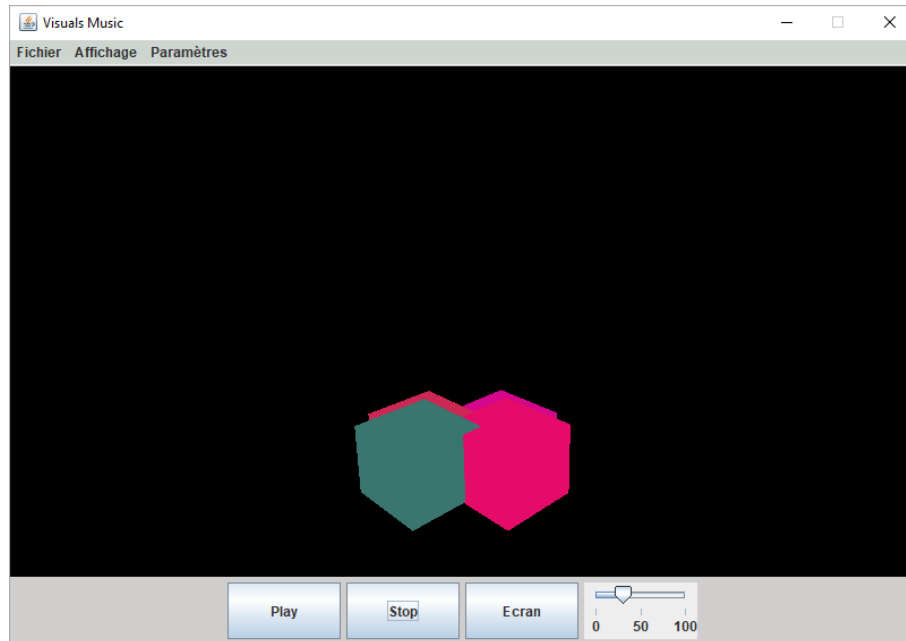


Illustration 7: L'interface graphique 3D où les cubes se coupent

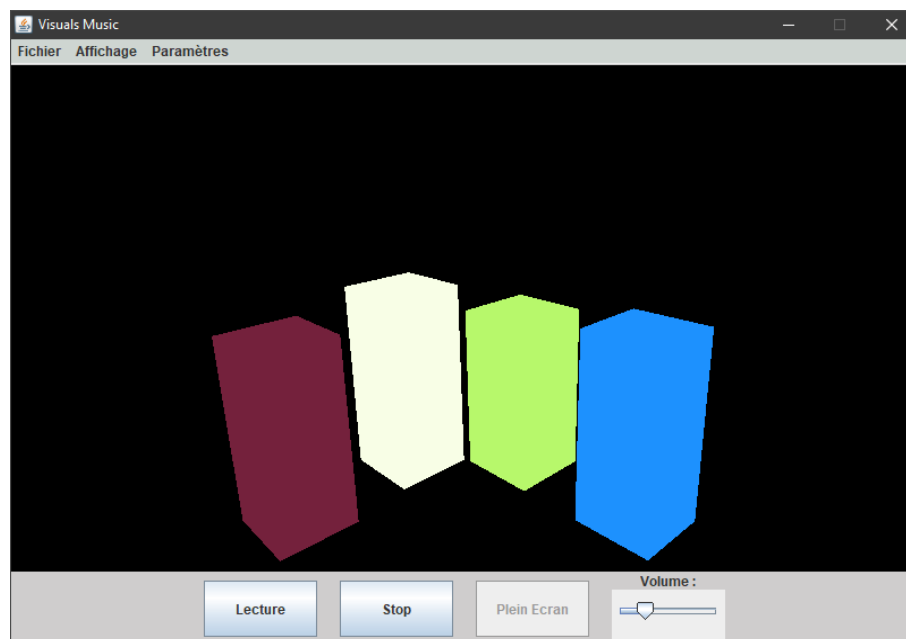


Illustration 8: L'interface graphique 3D à la fin du sprint 2

Pour la fin de notre sprint “trois”, avant le rendu final, nous avons revus la Javadoc afin d’avoir une langue unique et de faire correspondre la documentation au code. Nous avons aussi revus le code, afin de le nettoyer. Nous avons aussi réglé des problèmes - espacement qui décalait l’affichage 2D vers la droite - et désactivé certains éléments qui n’ont pas été proprement implémenté comme le bouton plein écran qui ne fonctionne pas.

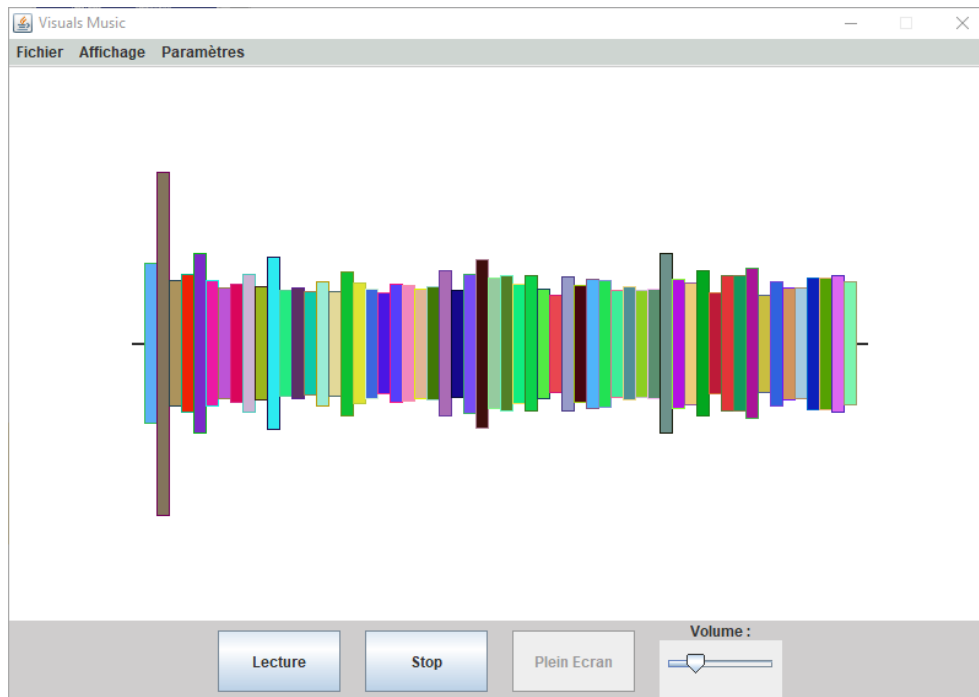


Illustration 9: L'interface graphique 2D à la fin du sprint 3

Résultats

A la fin de ce projet tuteuré numéro trois, nous avons obtenus une application Java permettant de lire et de visualiser en 2D et 3D de la musique provenant de fichier audio d’extension “.wav”. De plus notre application permet des interactions simples sur la lecture audio, comme par exemple la pause ou dé-pause de la lecture, ou encore son arrêt. On est également en mesure d’amplifier ou de réduire le volume de la chanson joué. L'utilisateur peut, s'il le souhaite, modifier l’affichage en donnant des couleurs aux dessins, une taille, un espacement, ou encore une amplitude. Pour finir, notre application, si un problème se pose, gère les erreurs en affichant un message.

Suite à des tests par des personnes externes au projet, nous avons pus tirer des points positif et/ou négatif de notre application.

Tableau 5: Avantages & inconvénients suivant plusieurs éléments de notre application

	Affichage 2D	Affichage 3D	Général
Avantage	<ul style="list-style-type: none"> • Possibilité de choisir les couleurs • Clair et simple • Sobre 	<ul style="list-style-type: none"> • Divertissant • Possibilité de choisir les couleurs 	<ul style="list-style-type: none"> • Pourrait être commercialisé
Inconvénients	<ul style="list-style-type: none"> • Manque de choix de forme 	<ul style="list-style-type: none"> • Manque de choix de forme • Manque de nombre de forme 	<ul style="list-style-type: none"> • Absence de la progression de la musique • Éléments désactivés • Boutons non élégant • Manque de boutons (suivant, précédent, ...)

Conclusion

Bilan du travail / Perspectives

Si notre projet était à refaire, nous commencerions notre projet dès le début du sprint “un”. Nous ferions un backlog dès le début du projet, afin de connaître son avancement et de connaître les tâches restantes. Nous essaierions d’être plus investis, afin que tout le monde puisse coder sur tous les sprints. Nous resterions dans le cadre des méthodes agiles, et respecterions les délais, pour ne pas prendre de retard sur le projet.

Pour continuer ce projet, nous pourrions l’implémenter sur mobile, ajouter des fonctionnalités (suivant, loop, précédent, random, ...) afin de correspondre à notre cahier des charges, créer de nouveaux visualiseurs afin d’avoir des affichages différents et originaux.



Illustration 10: Exemple de visualiseur

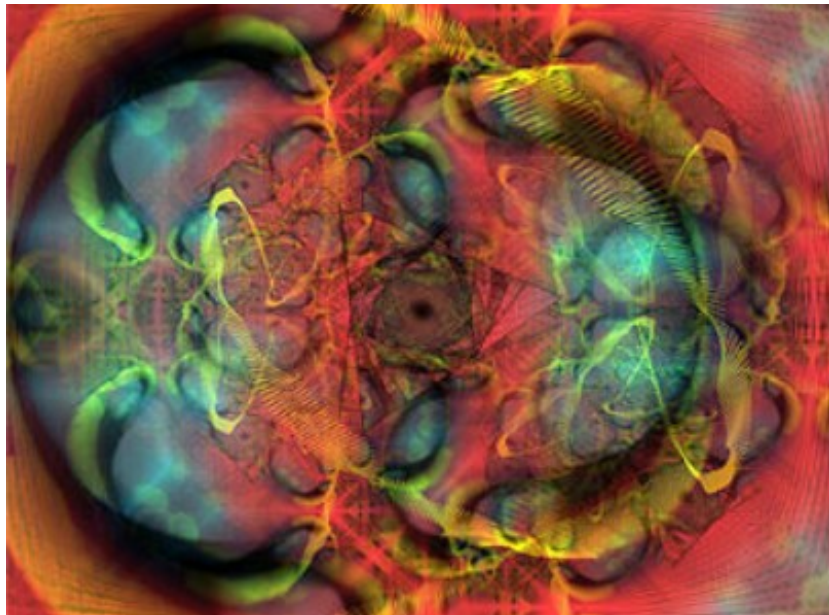


Illustration 11: Exemple de visualiseur

Bilan personnel

Notre projet nous a permis de travailler en groupe, tout en apprenant de nouvelle connaissance comme l'utilisation de JOGL. Nous nous sommes rendus compte, qu'on devait plus s'investir pour les sprints "deux" et "trois", car nous avons pris un peu de retard lors de la fin du sprint "un". Ce projet nous a aussi permit d'appliquer la structure MVC sur un projet conséquent de notre choix. Les méthodes agiles ont été très pratique lors de la durée de notre projet. Les outils étaient nombreux pour nous permettre de réussir, par exemple, pour savoir ce qu'il fallait faire nous pouvions utiliser un backlog. Ces méthodes nous ont permis de bien avancer et de mener à bien notre projet. Nous avons utilisé Git, ce qui nous a permis d'approfondir nos connaissance dans ce logiciel.

Annexe

What did we have to do?

The three of us had to develop an application for our PT3, entitled visualization of music that Ms. Boltcheva suggests. Therefore, Ms. Boltcheva is our tutor. She helped and guided us a lot thought our project. The purpose was to generate shapes in 2D and 3D while a music was played. For doing so, we had to developed a Java application using different software like: Eclipse, GitHub or also Discord. We started to develop the application in November, starting to create the early interface. We had about four mouths to familiarized with the subject, the programming language and complete this project. We, finally, decided to name our project “Visuals Music”.

We aim to entertain a young public (about 16-24) because the youngest are more prone to listen music and even more to listen it on their mobile phones. Of course, all can use the application because our graphic user interface is intuitive.

When our application will be complete, we will try to convert it for mobile use (smartphone) because usually Visuals Music is devoted to PC.

What did we do?

We have developed an application which permit to watch music. For doing so, we had to create 2D and 3D Visualizers and put them into a graphical interface.

Also, our application implements the general functionalities of a music player, which are:

- Play Button
- Pause Button
- Stop Button
- File Finder for open a file into the computer

Finally, we want to make more features, as:

- Next Button
- Previous Button
- Random Button
- Loop Button
- Volume Slider

- Progression Slider

Before starting programming, we had to define a framework to specify which elements will or will not belong to our project. As a conclusion, our application will not be published into a platform as App Store, Play Store or even Microsoft Store, it will not be linked within the Internet and it will be free of charge.

We also had constraints as lots of coming deadlines, unknown programming environment, agile methods which we had to apply. Indeed, agile approach was a constraint because we never apply it and we just learned them two weeks before the project started.

For the agile method, we choose to multiply three two-weeks sprints with one week to present our advancement to Ms. Boltcheva. Doing so we had be able to determine our next objectives. Unfortunately, Ms. Boltcheva was not always in Saint-Dié-des-Vosges when we had to present our project to her, so we made some videos available on:

<https://www.youtube.com/playlist?list=PLUav5QNakljAY9G2W74ax9nhDw7ve02k4>

How did we do it?

At the beginning we did a meeting between us for impose some rules (code commentary, good use of GitHub, no magical number in our application, ...), do the provisional schedule and the UML diagram.

The preliminary studies began and was composed by:

- | | |
|-------------------------------------|--------------------------|
| • Needs analysis | Océane |
| • Analysis of what currently exists | Stéphane, Arthur |
| • Benchmarking | Stéphane |
| • Preliminary schedule | Stéphane |
| • Specifications | Stéphane |
| • Guide interviews | Stéphane, Océane, Arthur |
| • Interviews | Stéphane, Océane, Arthur |
| • Slideshow | Stéphane, Océane, Arthur |
| • UML diagrams | Stéphane, Océane |

At the first sprint, we were already late, so we made the programming, supposed to last two weeks, and the presentation to our tutor in the same week. We started to create the graphic user interface and to think about how read a music file. Because we could not find a way to did it, we asked our tutor how we could. We implemented a button Pause and a button Play. We cannot read something else than .wav files yet. After, the graphic user interface created, we presented our project to Ms. Bolcheva, sprint “one” was finished. After sprint

“one”, we bound Pause and Play buttons to make one, and created a Stop button. Also we wrote the Javadoc (legit commentaries in Java) at the same time that we programmed. Stéphane did that that part.

During the second sprint, we made a backlog for prioritize all the tasks we had to do. Happily, we were on schedule. Arthur did the poster that was asked in Communication classes. And all together we started to write this report. For the IT part, we had to add a condition because the Stop button was not working. We asked help to Ms. Bolcheva to calculate a Fast Fourier Transformation (FFT) that we did not understand. We had to use a FFT to convert bytes into frequency. With the frequency we could displayed the music visualizers. As we implemented the 2D visualizer, we displayed it, but, sadly, we had a display issue. The display moved from right to left and the drawing repeated themselves. To resolve this issue, we had to draw over all the drawing a big white rectangle for clearing the screen. The first Stop button had a problem when we double clicked on it, we added a condition to have not this problem again. We asked help to our tutor to know how could we draw 3D rectangles. Sprint “two” was finished. Stéphane did also this part. Here again Stéphane wrote the Javadoc at the same time that he programmed.

For the third and last sprint, we were on schedule although we did not finished our project. Stéphane made the 3D Visualizer using JOGL, a Java library which extends GOGL, a C library which allow 3D shape drawing. He also made a Settings window allowing the users to change color, size, number of shapes on the screen. Océane programmed shortcuts for the keyboard (Space for pause/unpause, Echap for quit the application, ...). As it is for the Javadoc, Stéphane and Océane wrote it alongside they programmed. As the third sprint ended, Océane and Stéphane started the final report and the slideshow needed for the presentation on the 14th January and Stéphane also wrote the English slideshow.

What did we use?

We use technical tools to carry out our project as Discord, Google Drive, GitHub or Eclipse – Java.

Discord is an application which permit to chat with people online. We can use it to chat, share image or even small files. We used it to gather information that we could split it in different specifics channels. This was organized in groups such as “archivé” (archived) or also “temporaire” (temporary).

Google Drive is a platform where the users can share data and work all together at the same time on a same file. We used Google Drive to the shared files that we can write and read simultaneously. We used Google Drive to work on project text part like this report, or the preliminary study. Like that we were able to work all on a different part on every writing file and to do the job quicker.

GitHub is similar than Google Drive but more programmer-focused. Indeed, there is a version control system that permit, if we do something wrong, to go back at a version which was working. It was a useful tool, because we could work on different version and finally merged them avoiding a lot of problems.

Eclipse – Java is an Integrated Development Environment (IDE). An IDE is a tool for programmer. With that we were able to program with a text editor, to compile with a compiler and to test our application, debugging the program doing so.

What will we remember?

About us

The working atmosphere between us was great although we did have a lack of commitment at first. One of our greatest asset was our communication. Indeed, we fantastically used all our communication tools, doing so we were able to pass information rapidly and effectively.

About the project

First we apply specific methods as agile approach or even file-sharing software (Git) that brought us achievements. Also, we had a lot of deadlines due to agile methods. Because of that, we learned a lot more of project management.

If we had to it again:

- We would make a backlog earlier
- We would be more invested

Cahier des charges

Contexte et définition du projet

En France, près de 99% de la population écoute de la musique. Parmi ces personnes, se sont les 16 - 24 ans qui en écoutent le plus. C'est pourquoi il est nécessaire de toucher cette population. De plus les jeunes disposent de très peu de financement et favorisent donc les applications gratuites ou quasi-gratuites. Aussi ce sont eux qui sont les plus sur leur smartphones.

D'ailleurs le marché du smartphone est en pleine expansion. Par exemple, tous les ans, un nouvel iPhone sort. On estime, qu'actuellement, six milliards de smartphones sont présent sur le globe, soit une augmentation de 56% par rapport à 2017.

Nous considérons donc qu'il est important de produire une application mobile gratuite permettant de jouer de la musique.

Le Cahier des Charges ci-présent va donc porter sur le sujet suivant :

- Programmation d'une application mobile permettant de visualiser de la musique
 - Application mobile gratuite permettant de visualiser une musique que l'utilisateur choisira.
 - Analyse en temps réel de la musique et génération d'une sortie visuelle.

Objectif du projet

- Offrir une expérience utilisateur supérieure aux concurrents.
- Créer une application mobile donnant envie à l'utilisateur de l'utiliser.
- Lier aussi bien visualiseur de musique que lecteur de musique.
- Visualisation synchronisé avec la lecture de la musique.
- Créer une application mobile générant des formes en fonction de divers éléments de la musique (timbre, fréquence, ...).

- Créer une application gratuite.
- Combiner un maximum de point forts des applications concurrentes.
- Éviter un maximum de point faibles des applications concurrentes.
- Donner à l'utilisateur la possibilité de pouvoir choisir la musique qu'il souhaite.
- Bonne lecture des fichiers audio (sans saturation).
- Projet innovant.

Périmètre du projet

- L'application ne sera pas relié à internet.
- L'application ne sera pas payante.
- L'application ne sera pas publiée. (Google Play, Play Store, ...)

Contraintes

- Temps limité.
- Environnement de travail inconnu (zéro programmation pour application mobile).
- Public assez restreint.
- Application des méthodes agiles.
 - Trois sprints de deux semaines.
 - Une semaine pour présenter au tuteur du projet suite à chaque sprint.

Description fonctionnelle des besoins

- Attendus :

- Lecture de musique
- Affichage de formes 2D et 3D relatives à la musique
- Analyse en temps réel du morceau musical
- Choix de n'importe quelle musique disponible sur l'appareil mobile
- Ergonomie un minimum poussée
 - Boutons visibles et explicites pour tout type d'utilisateur
- Bouton
 - Play / Pause
 - Stop
- Apprécies :
 - Affichage titre de la musique
 - Possibilité d'écouter la radio
 - Bouton
 - Suivant
 - Précédent
 - Lecture automatique
 - Lecture en boucle
 - Accéder à l'auteur
 - Accéder à l'album
 - Plein écran

- Volume
- Playlist
 - Pouvoir en créer
 - Pouvoir les sauvegarder
- Affichage des paroles
- Possibilité de jouer des musiques à la suite (file d'attente)

Diagrammes UML

Les diagrammes ci-dessous sont les dernières version du 13 janvier 2020.

Cas d'utilisation

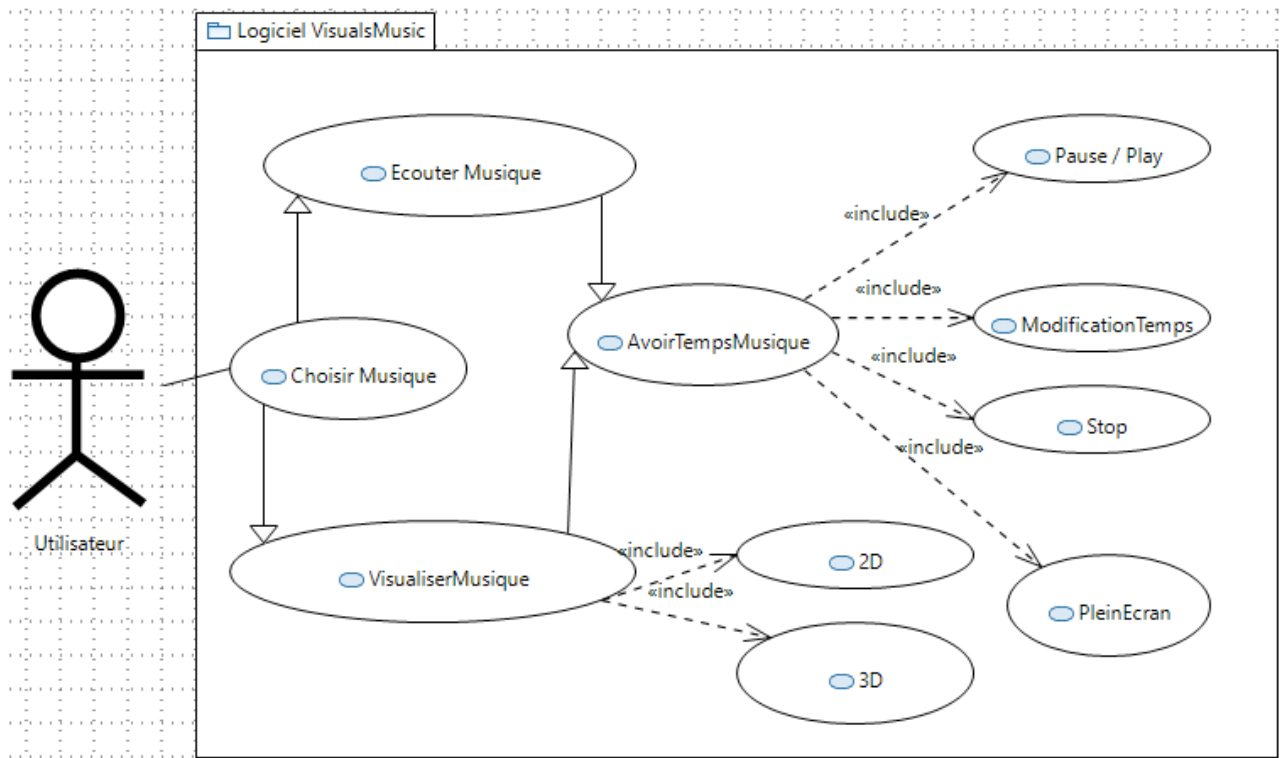


Diagramme 1: Cas d'utilisation de l'application Visuals Music

Classe

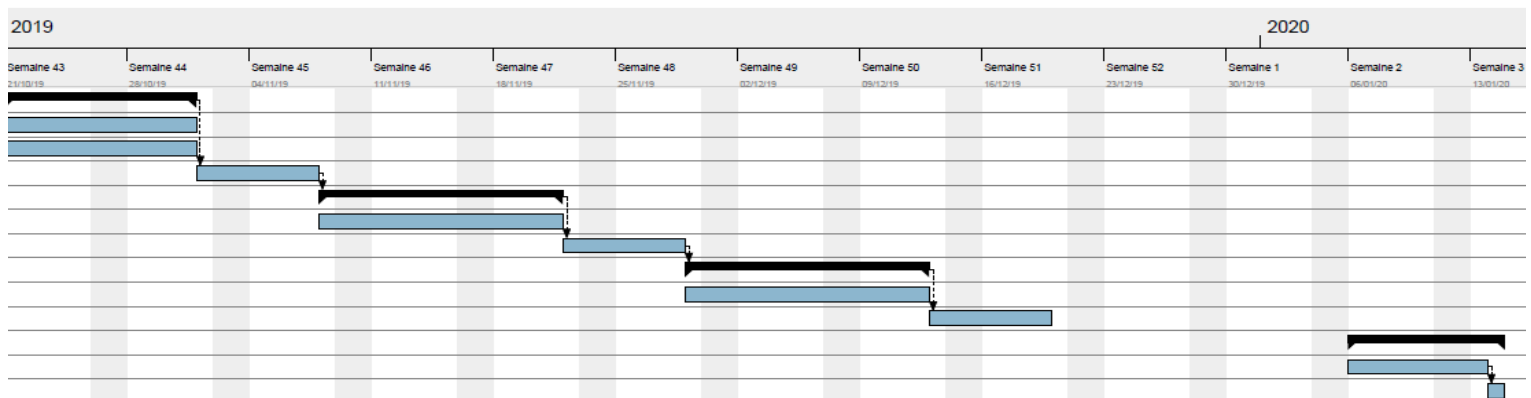
Le diagramme de classe étant trop important pour être mis sur un page A4, celui-ci a été limité aux noms des classes et aux relations les liant.

Planning

Préalable

GANTT project			
Nom		Date de déb..	Date de fin
☐ • Sprint 1		21/10/19	31/10/19
• Fabrication de l'Interface Graphique		21/10/19	31/10/19
• Mise en place d'algorithme simple		21/10/19	31/10/19
• Présentation au tuteur 1		01/11/19	07/11/19
☐ • Sprint 2		08/11/19	21/11/19
• Tâches définies au sprint précédent		08/11/19	21/11/19
• Présentation au tuteur 2		22/11/19	28/11/19
☐ • Sprint 3		29/11/19	12/12/19
• Copie_Tâches définies au sprint précédent		29/11/19	12/12/19
• Présentation au tuteur 3		13/12/19	19/12/19
☐ • Soutenance		06/01/20	14/01/20
• Préparation Soutenance		06/01/20	13/01/20
• Présentation		14/01/20	14/01/20

Planning 1: Liste des tâches

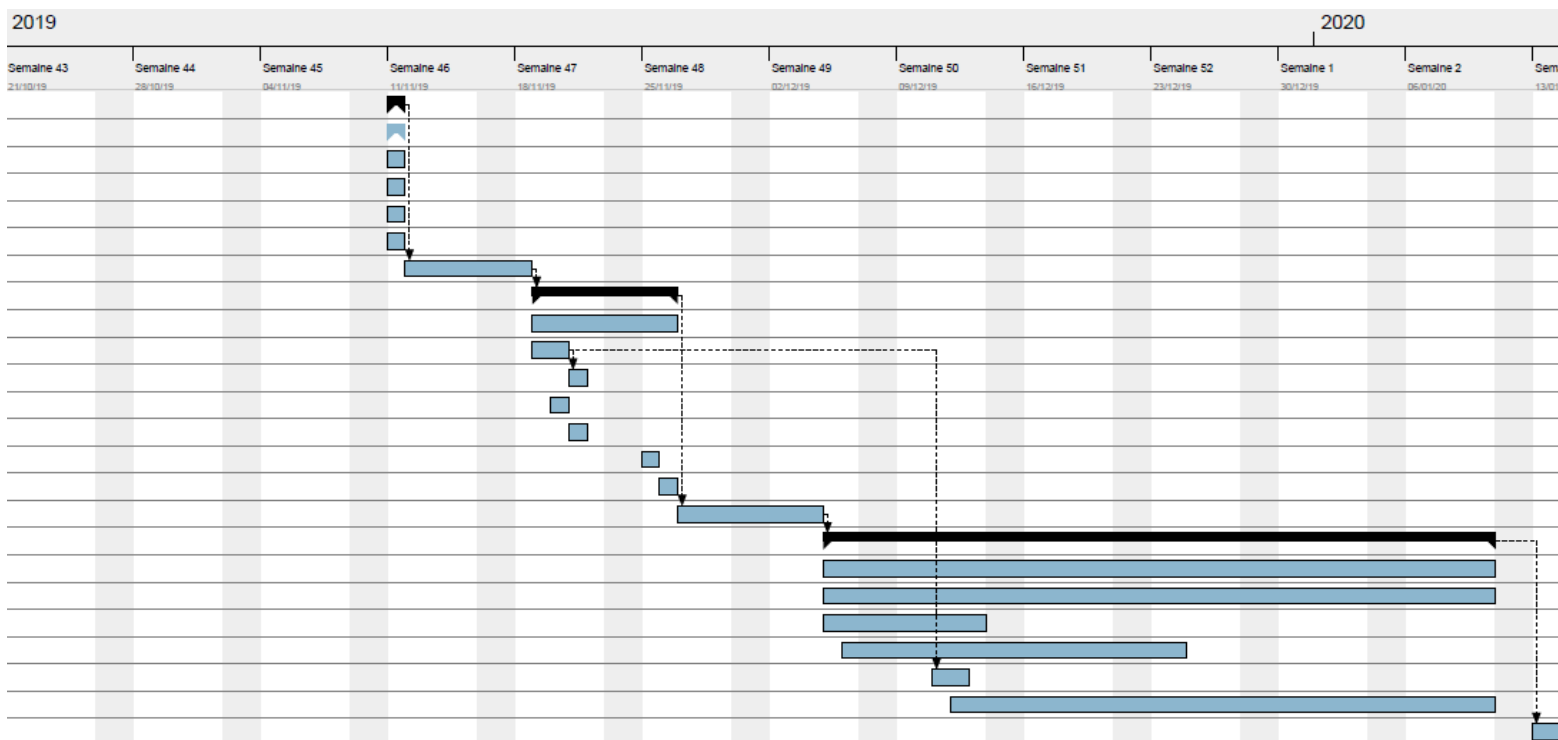


Planning 2: Durée des tâches

Final

GANTT project		
Nom	Date de déb.	Date de fin
• Sprint 1	11/11/19	11/11/19
• Fabrication de l'Interface Graphique	11/11/19	11/11/19
• FileChooser	11/11/19	11/11/19
• Bouton Pause	11/11/19	11/11/19
• Bouton Lecture	11/11/19	11/11/19
• Lecture d'un fichier audio	11/11/19	11/11/19
• Présentation au tuteur 1 (12/11/19)	12/11/19	18/11/19
• Sprint 2	19/11/19	26/11/19
• Correction erreurs	19/11/19	26/11/19
• FFT	19/11/19	20/11/19
• Visualiseur 2D	21/11/19	21/11/19
• Ajout plein écran	20/11/19	20/11/19
• Fusion Play/Pause	21/11/19	21/11/19
• Suppression code inutile	25/11/19	25/11/19
• Javadoc	26/11/19	26/11/19
• Présentation au tuteur 2 (26/11/19)	27/11/19	04/12/19
• Sprint 3	05/12/19	10/01/20
• Javadoc	05/12/19	10/01/20
• Correction erreurs	05/12/19	10/01/20
• Refonte Visualiseur 2D	05/12/19	13/12/19
• Paramètres	06/12/19	24/12/19
• Visualiseur 3D	11/12/19	12/12/19
• Slider Volume	12/12/19	10/01/20
• Présentation au tuteur 3 (13/01/20)	13/01/20	14/01/20

Planning 3: Liste des tâches



Planning 4: Durée des tâches