

Équipe Studio

Amice, Gabriel
(111 190 931)

Frigon, Alexandre
(111 175 404)

Rouleau, Alexandre
(111 186 058)

Trépanier, Olivier
(111 161 943)

Génie logiciel orienté objet
GLO-2004

Livrable 4

Travail présenté à
M. Jonathan Gaudreault

Département d'informatique et de génie logiciel
Université Laval
Automne 2017

Table des matières

Énoncé de vision	4
Introduction	4
Positionnement	4
Opportunité d'affaire	4
Énoncé de problème	4
Position sur le marché	4
Étude d'option	4
Description des intervenants et utilisateurs	5
Utilisateur ciblé	5
Liste des intervenants	5
Liste des utilisateurs	5
Objectifs et problèmes principaux des intervenants	5
Objectifs des utilisateurs	6
Environnement utilisateur	6
Présentation du produit	6
Description du produit	6
Sommaire des bénéfices	7
Dépendance du produit	7
Coût et prix	8
Installation	8
Sommaire des fonctionnalités du système	8
Autres exigences et contraintes	8
Diagrammes de boucles en direct	9
Diagramme de classe	9
Window	9
GaudrophoneController	9
SequencerManager	9
LiveLoopRecorder	10
Sequencer	10
LiveLoop	10
Song	10
PlayableChord	10
Diagramme d'états	10
Diagrammes de séquence	11
Saisies d'écran du produit fin	13
Conclusion	15



Contribution des membres de l'équipe

- Alexandre Rouleau :
Graphique de l'enveloppe (répondant aux exigences de M. Frigon). Ajout de la forme d'onde maître. Ajout et retrait de points. Insertion de formes pré faites avec l'interface (terminé par Frigon). Affichage des masques de couleurs différentes pour de nouveaux modes (tel que les touches présente dans une chanson). BEAUCOUP de bug fixes, trop pour énumérer. Sans mentionner l'aide fournis à Frigon pour de nombreuses fonctionnalités. Cela inclut la lecture de fichiers textes (implémentations de paragraphes et paragraphes de tailles variables), l'interface graphique, le métronome, l'insertion de touches via GUI, son calcul de polygon (amélioration) et autres. Oh, les diagrammes des boucles sont également par moi.
- Alexandre Frigon :
Pour ma part j'ai commencé par faire le module de lecture de fichiers au format imposé, ensuite j'ai fait l'affichage d'une fonctionnalité que nous appelons Jodrofun Hero. Il s'agit d'un tapis qui défile la chanson du fichier texte. Cette fonctionnalité sert d'affichage du fichier texte. J'ai aussi fait la majorité des changements en rapport à l'interface graphique et la gestion d'événement grâce aux delegates. J'ai aussi fait la deuxième version des boucles en direct pour prendre en charge les touches simultanées. J'ai aussi implémenté le métronome dynamiquement réglable. Aussi, j'ai collaboré avec Alexandre sur la panneau d'affichage d'insertion de forme en finalisant l'interface et en concevant l'algorithme de calcul de polygon. De plus, j'ai collaboré avec Olivier pour lier ses fonctionnalités à l'interface graphique tel que la gestion de contrôleur midi.
- Olivier Trépanier :
Correction de l'enveloppe sonore pour enlever les "clics" causés par le relâchement d'une note, gestion des inputs MIDI provenant de contrôleur extérieur. Première version du séquenceur et participation à la première version de la lecture du fichier texte ainsi que quelques bug fixes.
- Gabriel Amice :
Implémentation du premier prototype de Live Loops en collaboration avec Olivier. Quelques bug fixes.

Énoncé de vision

Introduction

L'application que nous avons en tête est un outil de création d'instrument de musique où il est possible de modifier chaque note visuellement et auditivement. Le but principal est de permettre aux musiciens de ce monde de réunir des notes de divers instruments au sein d'un seul et de pouvoir apprendre, jouer et performer avec celui-ci. Un mode d'apprentissage, de lecture de pièce et de "Live looping" seront également disponibles dans le produit.

Positionnement

Opportunité d'affaire

Les applications musicales actuellement sur le marché n'offrent pas la possibilité aux musiciens de créer leurs propres instruments à partir de sons personnalisés et de clip

audio comme le Gaudrophone le ferait. Cette fonctionnalité accompagnée bien-sûr de plusieurs autres tel que le “Live Looping” et le mode d’apprentissage rendront les musiciens heureux d’utiliser le produit.

Énoncé de problème

Un musicien classique ne peut jouer que d’un seul instrument à la fois. Avec le Gaudrophone, nous tentons de mettre fin à ce problème en donnant accès à plusieurs instruments dans un seul. En disposant des sons, l’utilisateur créera sa propre expérience unique avec l’ensemble de sons dont il a besoin pour jouer sa pièce.

Position sur le marché

Notre vision est unique en son genre. Les musiciens, ayant souvent l’esprit créatif aiguisé, seront comblé avec la multitude d’option de personnalisation qu’offre notre Gaudrophone. Variant de modification visuelle à modification auditive, l’application se démarquera sur le marché et sera la seule offrant autant de possibilité.

Étude d’option

Option	Avantages	Inconvénients
Instrument réel	<ul style="list-style-type: none">• Offre un son authentique.• Rapproche le musicien de son public	<ul style="list-style-type: none">• N’offrent pas la possibilité de jouer de plusieurs instruments à la fois.• Nécessite un instrument coûteux.• Nécessite du talent.
Application de création musicale (FL Studio, Ableton)	<ul style="list-style-type: none">• Offre un multitude d’option pour créer pratiquement tous les sons imaginable.• Offre des fonctionnalités très avancés.	<ul style="list-style-type: none">• Complexe et difficile d’utilisation.• Ne permet pas de jouer de l’instrument en direct.
Application de performance musicale (Ableton live lite, Xpand!2)	<ul style="list-style-type: none">• Permet de jouer en direct• Possibilité de connecter un contrôleur MIDI (piano) et jouer d’un instrument physique.	<ul style="list-style-type: none">• N’offrent pas la possibilité de jouer de plusieurs instruments à la fois.

Description des intervenants et utilisateurs

Utilisateur ciblé

Le Gaudrophone vise tout le monde ayant une passion pour la musique ou un intérêt à composer, apprendre et jouer des chansons. Peu importe l’âge, le sexe ou la provenance de l’utilisateur, l’application lui sera accessible dès qu’il aura les capacités d’utiliser un ordinateur.

Liste des intervenants

Nom	Description	Responsabilités
Gabriel Amice Alexandre Frigon	Programmeur analyste	<ul style="list-style-type: none">• Analyser les besoins du client.• Conceptualiser un système

Alexandre Rouleau Olivier Trépanier		<ul style="list-style-type: none"> répondant aux besoins. Développer le produit résultant de l'analyse et de la conception s'il y a lieu.
Jonathan Gaudreault	Client, expert du problème	<ul style="list-style-type: none"> Exprimer ses besoins de façon clair à l'équipe de développeur. Faire un retour sur les livrables de l'application avec une critique constructive pour remettre l'équipe dans la bonne direction si nécessaire.

Liste des utilisateurs

Nom	Description	Responsabilités
Musicien	Une personne ayant un intérêt pour la musique.	<ul style="list-style-type: none"> Faire l'utilisation des fonctionnalités offerte par l'application.

Objectifs et problèmes principaux des intervenants

Les intervenants souhaitent fournir un produit de qualité répondant aux besoins du client, l'analyse de ces besoins nous ont permis d'identifier les objectif suivant:

Objectif	Priorité (1>2)	Problèmes potentiels	Solutions actuelles
Jouer d'un instrument	1	<ul style="list-style-type: none"> Les sons pourraient ne pas être juste. il pourrait y avoir un délai trop important pour rendre l'expérience agréable. 	Depuis toujours les musiciens utilisent des instrument physique.
Personnaliser son expérience	2	<ul style="list-style-type: none"> Les formes créer pourraient être superposé et empêcher l'utilisateur de jouer de l'instrument. La couleur de la forme pourrait rendre le texte illisible. 	Aucune.
Enregistrer son progrès	5	<ul style="list-style-type: none"> L'utilisateur pourrait se décourager et ne plus jamais utiliser l'application si ses données sont corrompues. 	Enregistrement audio.
Apprendre à jouer	4	<ul style="list-style-type: none"> L'apprenti musicien pourrait se décourager si il y a des termes trop compliqués ou si c'est trop difficile. 	Cours, essaie-erreur.
Performer à plusieurs instruments	3	<ul style="list-style-type: none"> Puisque ce mode sera utilisé en performance en directe il n'y a pas de place à l'erreur et les fonctionnalité doivent être facile d'accès. 	Les gens se regroupent pour jouer plusieurs instrument en même temps.

Objectifs des utilisateurs

Comblent son esprit musical en s'amusant avec l'application, en apprenant comment jouer de l'instrument, en performant en direct avec les fonctionnalités de "Live Looping" et en crée ses propre instruments. Le principal but du Gaudrophone est de divertir la personne qui l'utilise.

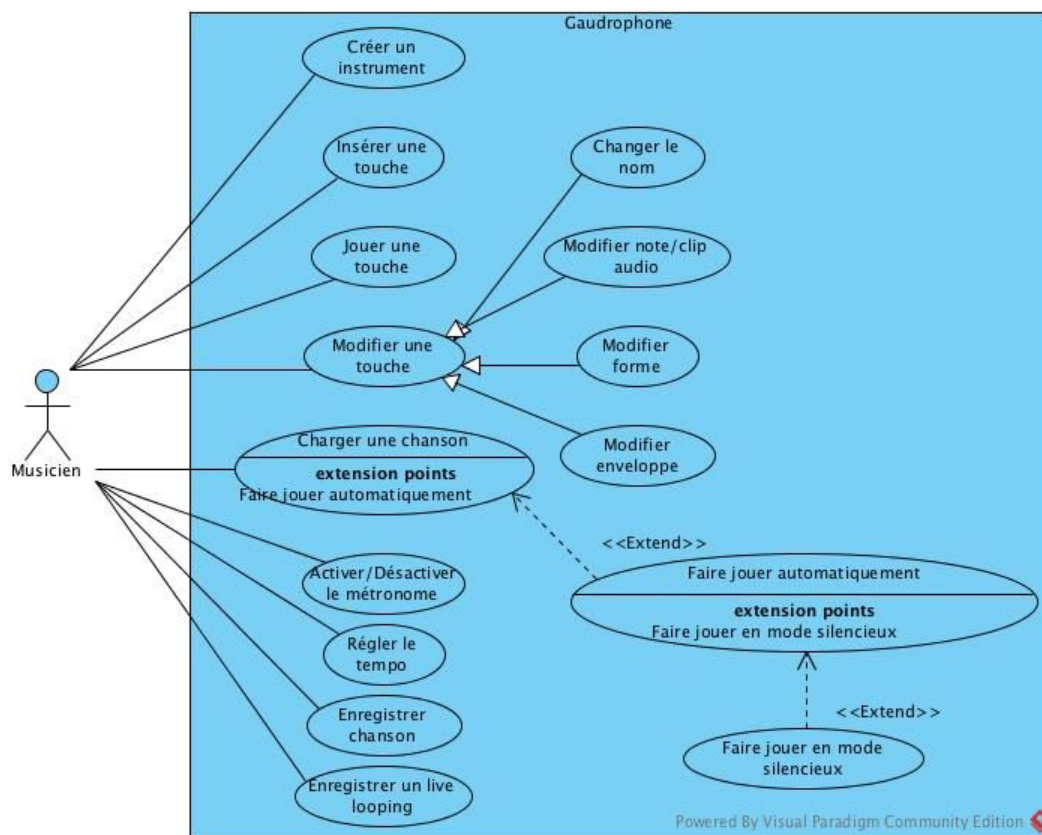
Environnement utilisateur

L'application aura un thème sombre puisqu'elle sera utilisé en public lors des spectacle que feront les musiciens et devra donc être subtile. La majorité du temps l'utilisation du logiciel se fera dans une maison, un bureau ou une place publique équipé d'un accès internet. Il n'y aura d'autre restriction que le thème de l'application puisque l'environnement ne posera généralement pas de problème au bon fonctionnement du logiciel.

Présentation du produit

Description du produit

L'application est utilisée sur un ordinateur personnel et offre une liste de fonctionnalités permettant de faire toute sorte de création, d'apprentissage et de performance musicale tel qu'illustré dans le diagramme suivant:



Sommaire des bénéfices

Fonctionnalité sollicitée	Bénéfices
---------------------------	-----------

Modification visuelle	Permet au musicien de créer une expérience unique et de se repérer facilement lorsqu'il fera des performance en direct.
Modification auditive	Offre la possibilité aux musiciens de s'exprimer à leur façon et de créer des pièces riches et agréable pour tous.
Mode "Live Looping"	Avec le "Live Looping", l'utilisateur se verra transformer en un groupe à lui seul. La fonctionnalité permet d'enregistrer des boucles qui joueront toutes en même temps donnant l'illusion que plusieurs personnes donne un spectacle.
Mode d'apprentissage	Ce mode ouvre l'accès à l'application à un plus grand public en enseignant des chanson aux utilisateurs ayant un peu moins d'expérience dans le domaine musical.

Dépendance du produit

Puisque le produit est programmé avec le langage "Java" il est crucial qu'une version compatible soit installé sur le terminal de l'utilisateur pour assurer le fonctionnement normal du produit.

Coût et prix

Les outils de développement utilisés sont gratuits et le temps consacré au projet n'est pas rémunéré, donc il n'y a aucun coût lié au développement. L'application sera évaluée comme projet scolaire et ne sera probablement pas en vente.

Installation

L'installation sera faite par un programme d'installation que le client obtiendra sur le web en assumant que le logiciel "Java" est déjà installé sur le poste de l'utilisateur.

Sommaire des fonctionnalités du système

- Créer un instrument
- Sauvegarder un instrument
- Insérer une touche
- Modifier une touche
 - Changer le nom
 - Modifier le son
 - Charger un fichier audio
 - Créer un son
 - Modifier son enveloppe
 - Modifier Attaque
 - Modifier Decay
 - Modifier Substain
 - Modifier Release
 - Modifier l'apparence
 - Changer la forme
 - Ajouter un point
 - Déplacer un point
 - Retirer un point
 - Changer de couleurs
 - La forme
 - La forme appuyé
 - Les bordures individuellement

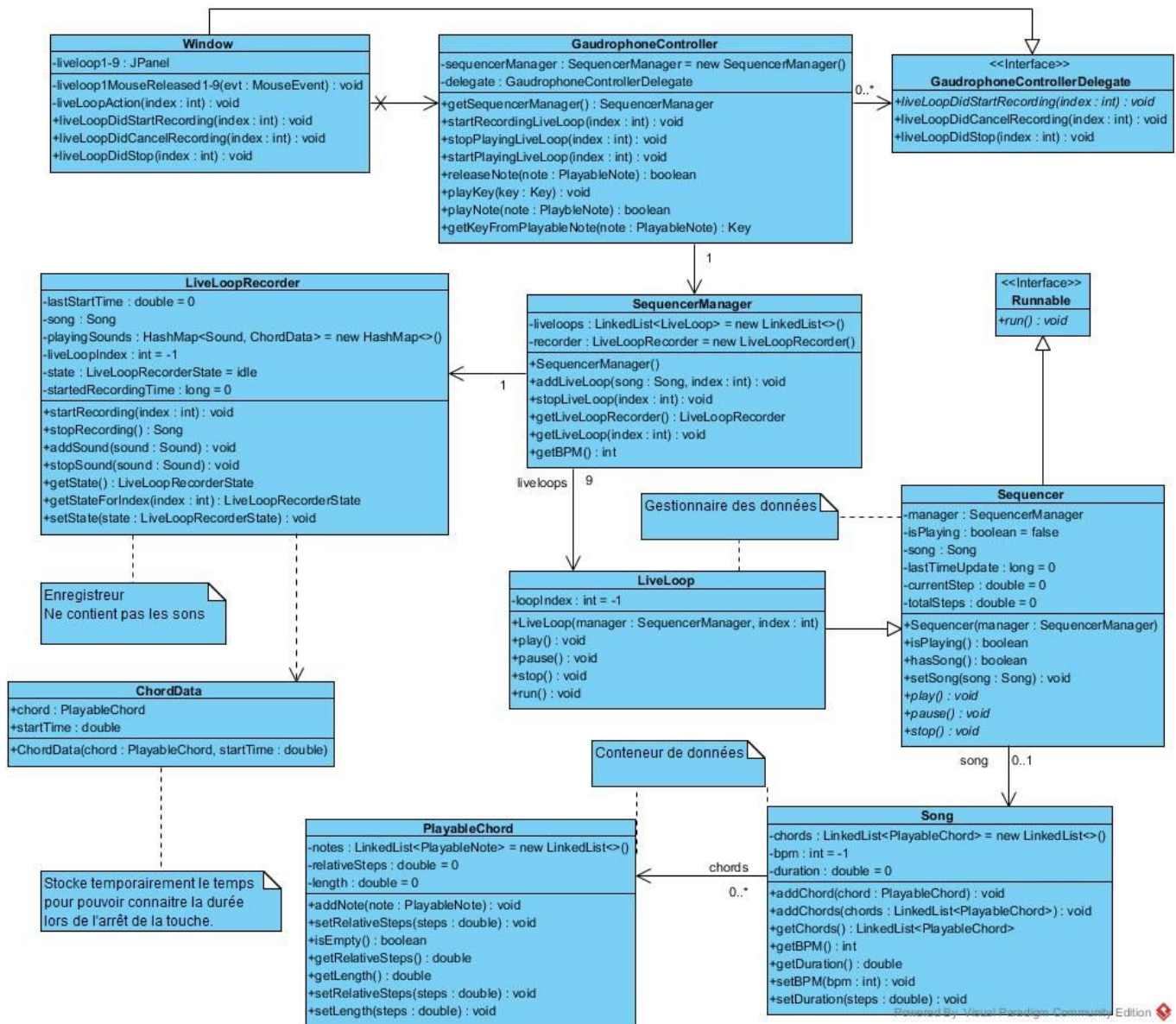
- Changer la largeur des bordures
- Jouer une touche
- Charger une chanson
 - Faire jouer une chanson automatiquement
 - Faire jouer une chanson en mode silencieux
- Utiliser le métronome
- Régler le tempo
- Activer une boucle en direct
 - Enregistrer une boucle
 - Faire jouer/arrêter une boucle

Autres exigences et contraintes

- L'application devra être terminée pour la remise finale du livrable 4, le 17 décembre 2017 avant 13h.
- L'application doit être développée en Java, plus précisément avec Swing et l'IDE de NetBeans

Diagrammes de boucles en direct

Diagramme de classe



Window

Window est le JFrame de l'application. Il envoie les inputs en passant par le contrôleur. Il peut recevoir des événements avec le Delegate (observateur) du contrôleur.

GaudrophoneController

Contrôleur fait le lien entre l'interface et le séquenceur. Il envoie les inputs vers le séquenceur ou l'enregistreur.

SequencerManager

SequencerManager possède plusieurs séquenceur : 9 boucles et un joueur de chanson. Il gère aussi l'enregistreur et attribue les objets aux bonnes boucles.

LiveLoopRecorder

LiveLoopRecorder est la classe qui s'occupe d'enregistrer les touches appuyées par l'utilisateur. Il le fait à l'aide d'une map de son et de la structure ChordData (un accord d'une note et un temps de départ) afin de déterminer la durée d'une touche. À la fin de l'enregistrement, il retourne un objet Song.

Sequencer

Sequencer implémente Runnable et est le parent des objets permettant de lire l'objet Song.

LiveLoop

LiveLoop hérite de Sequencer. Il convertit l'objet Song en une boucle et s'arrête lorsque demandé.

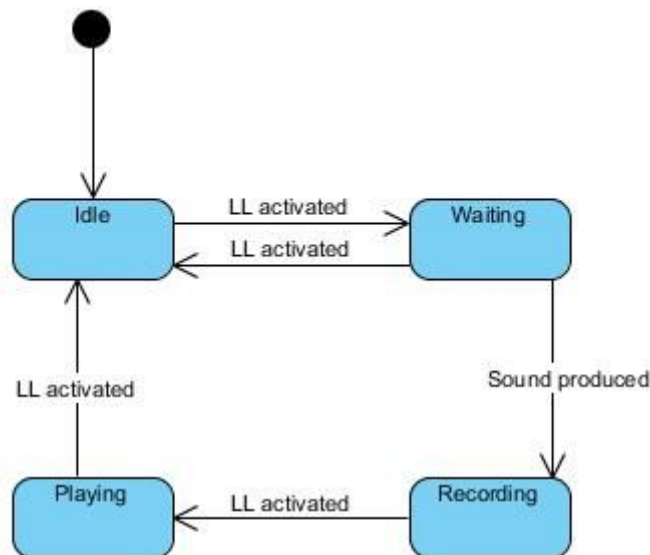
Song

Song est composé de plusieurs accords, d'un BPM et d'une durée.

PlayableChord

PlayableChord est LA classe qui contient les données pour jouer un son. Il contient une liste de PlayableNote, qui est principalement une fréquence, pour déterminer les notes qui sont a jouer dans cet accord. Il contient aussi le moment que cet accord devrait être joué ainsi que sa durée. Avec ses informations, les séquenceurs peuvent faire jouer les touches appropriées.

Diagramme d'états

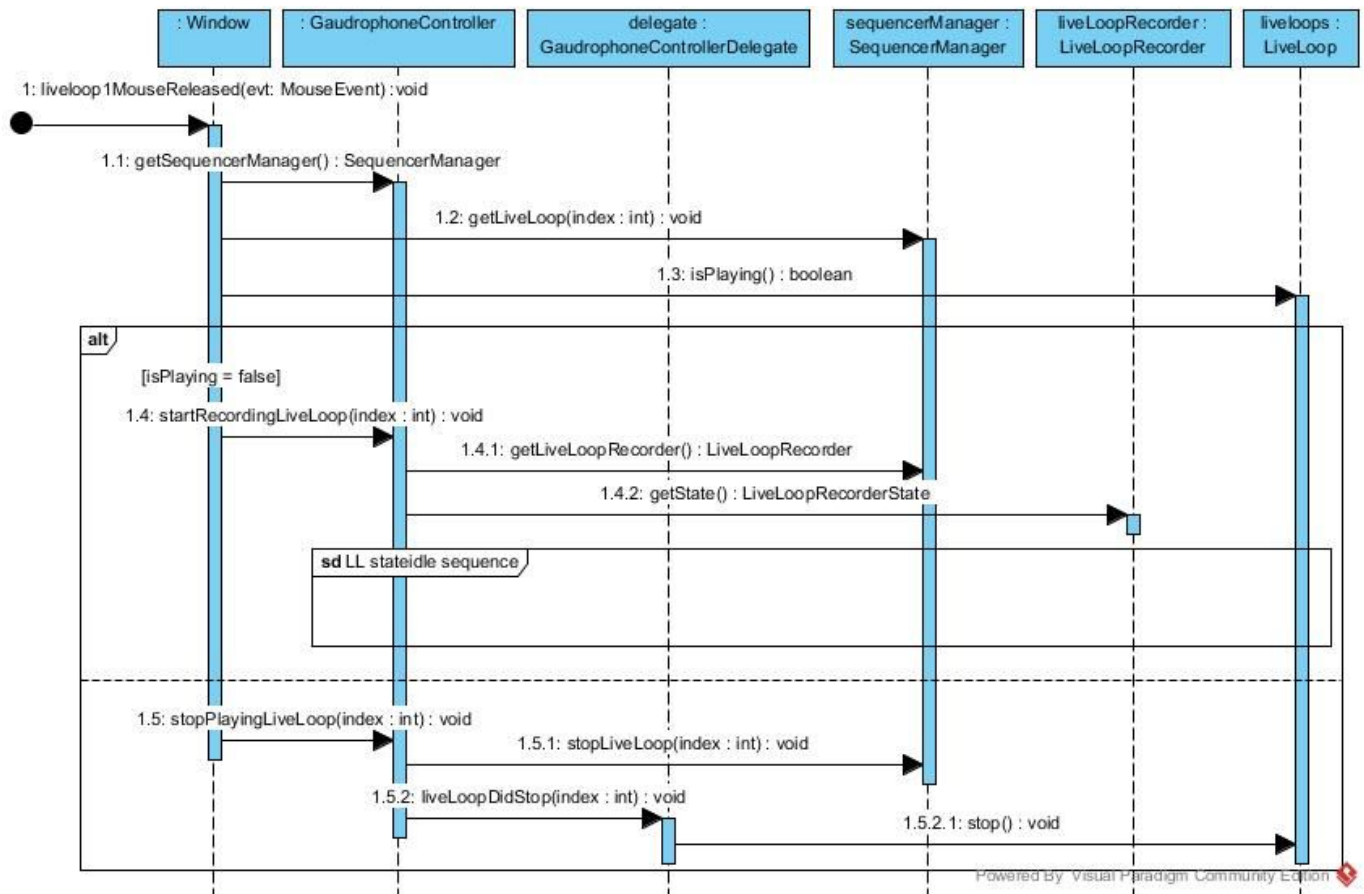


- Activate once, wait for input
- If activated again before input, go back to sleep
- If input, start recording inputs until activated second time
- Play in loop
- Activated a third time : go back to starting point

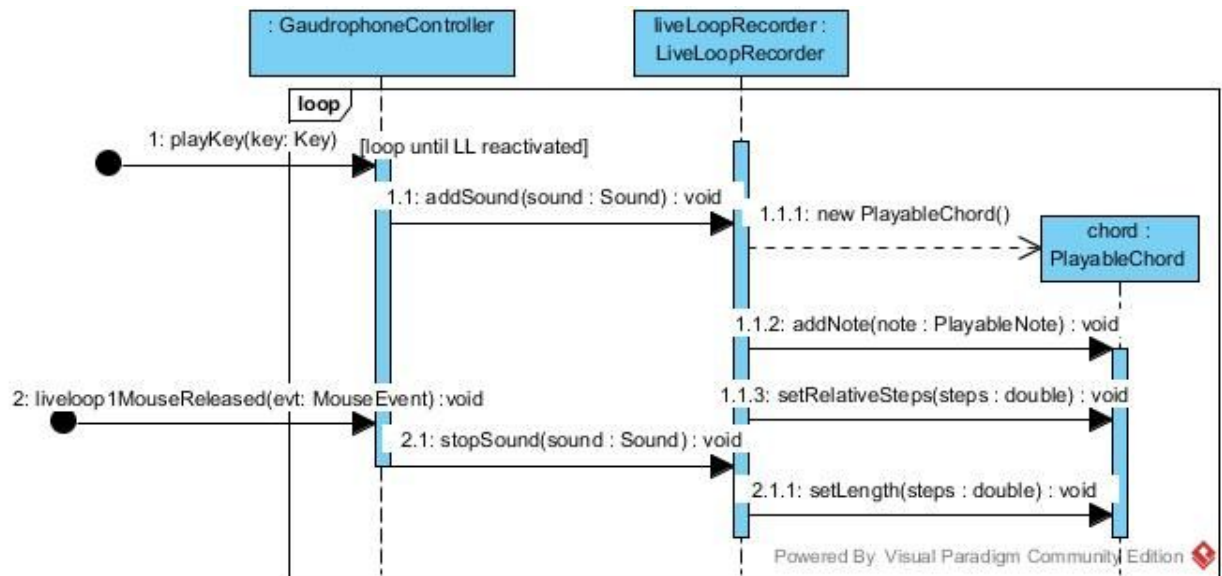
Powered By Visual Paradigm Community Edition

Au départ, toutes les boucles sont inactives. Lorsqu'une boucle est activée une première fois, il passe en mode attente. Si l'utilisateur appuie encore sur la boucle, l'enregistrement ne s'effectue pas et retourne en mode inactive. Or, s'il y a une touche appuyée, il passe en mode enregistrement. Toutes les touches sont ensuite enregistrées jusqu'à ce que la boucle soit appuyée. Elle passe alors en mode jeu et reproduit les sons. Lorsqu'elle est appuyée une troisième fois, elle arrête de jouer et devient inactive.

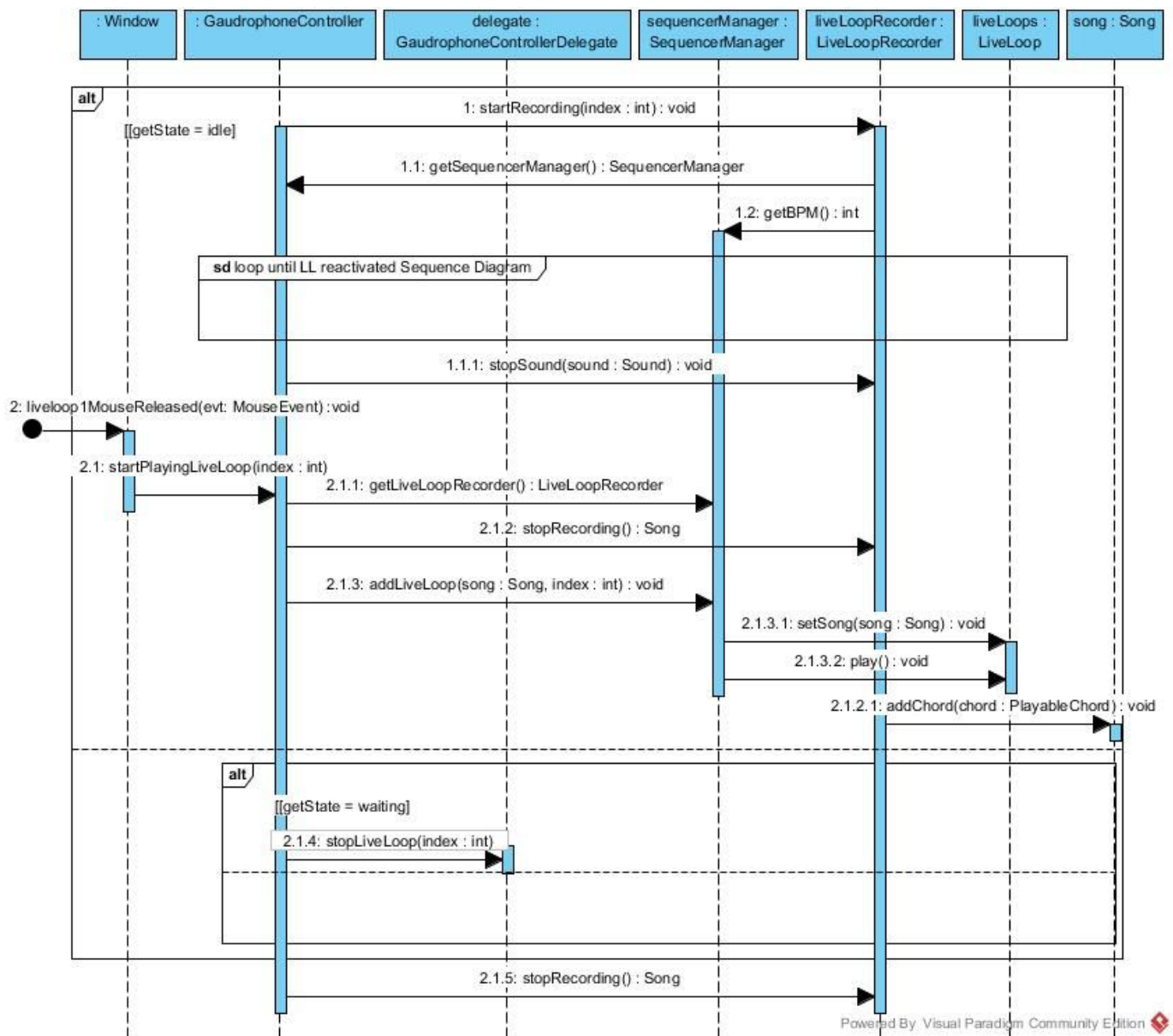
Diagrammes de séquence



La création d'une boucle est assez simple : l'utilisateur appuie sur le clavier ou sur l'endroit réservé à cet effet pour activer l'enregistrement, tape sur des touches, appuie une seconde fois sur la boucle pour terminer son enregistrement. La boucle se met à jouer jusqu'à une troisième action sur la boucle. Cela se fait via les events capté par Window, qui renvoie l'index de la boucle à une fonction privé qui change son apparence et qui envoie la requête au contrôleur (startRecording, startPlaying ou stopPlaying). Le contrôleur valide avec le LiveLoopRecorder s'il en déjà en train d'enregistrer ou s'il est libre. S'il est en train d'enregistrer, alors il arrête l'anciens enregistrement et démarre un nouveau. Par la suite, l'enregistreur devient en mode attente. Il attend une note avant de commencer son timer.

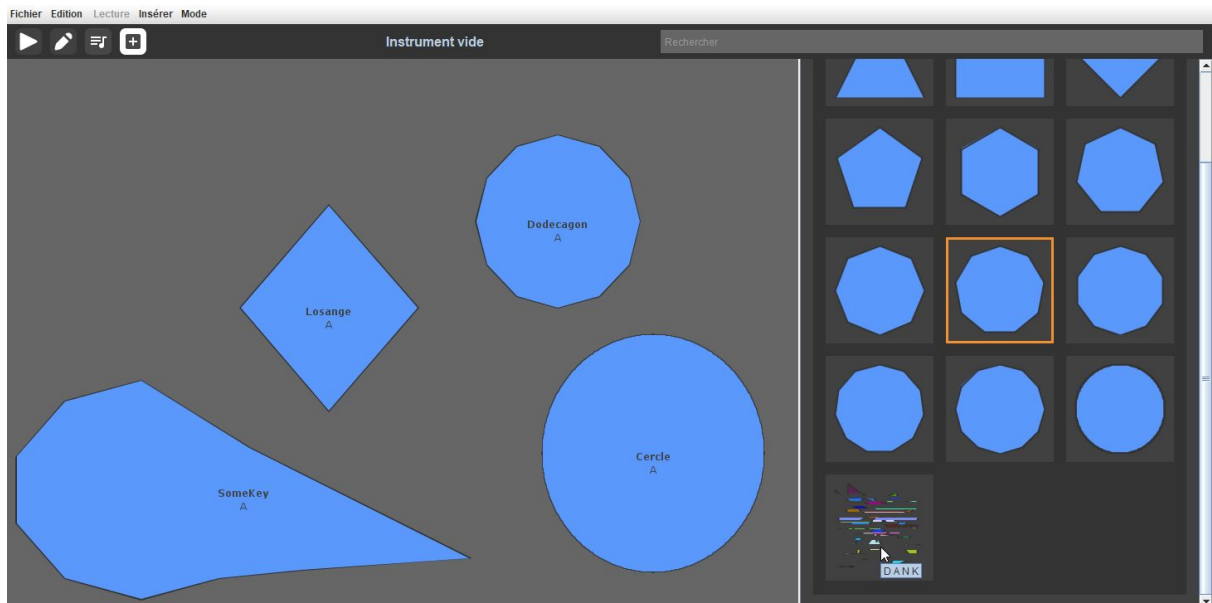


Une fois une touche appuyée, il passe en mode enregistrement et enregistre toutes touches appuyées. Pour ce faire, il utilise une map de ChordData, correspondant au son mappé et le temps de départ, afin de trouver la durée sur laquelle la touche a été appuyée. Une PlayableChord, qui représente un accord d'une note est créé.



Une fois que l'enregistrement est terminé par l'utilisateur, le contrôleur demande l'arrêt de l'enregistrement, récupère l'objet Song (composé principalement d'une liste de PlayableChord) et demande au séquenceur de démarrer la boucle. Le séquenceur attribue l'objet Song à la bonne boucle et la démarre dans un Thread.

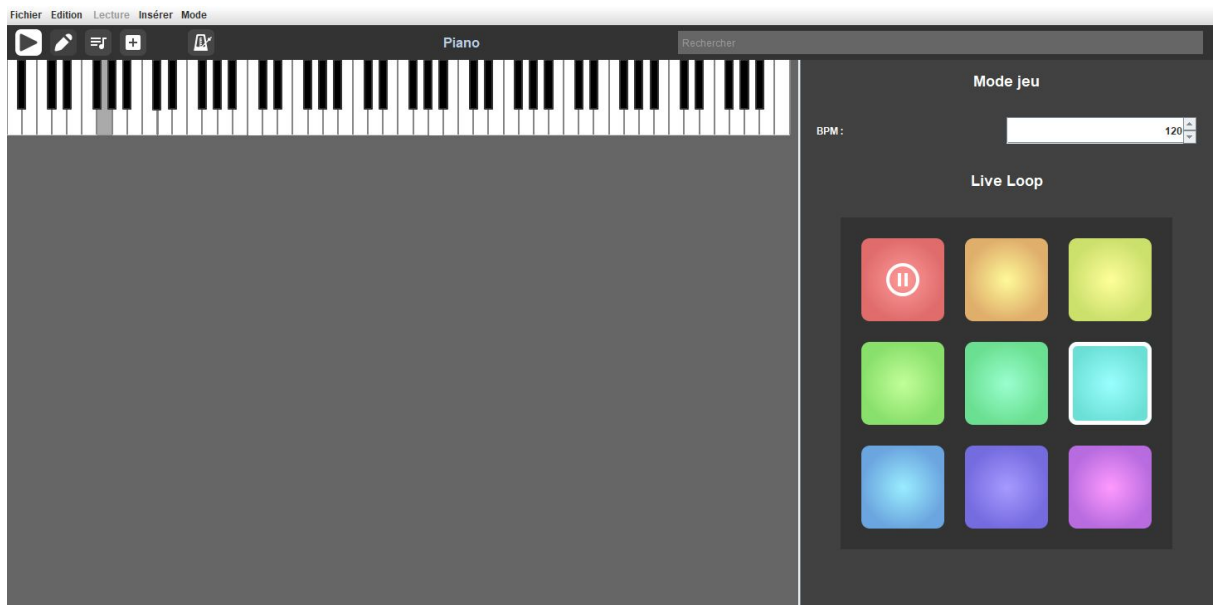
Saisies d'écran du produit fin



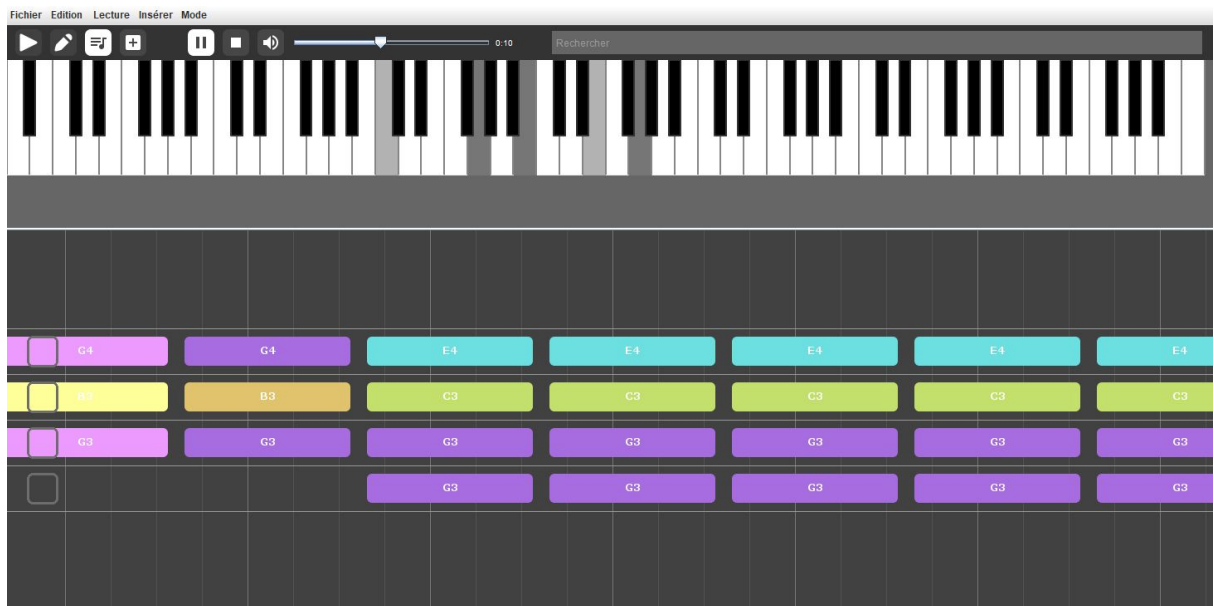
Insertion de nouvelles touches dans un instrument vide.



Modification du son d'une touche dans un piano.



Jeu libre avec un piano et boucles en direct.



Lecture d'une chanson avec un piano en mode lecture automatique.

Conclusion

Notre Gaudrophone possède de nombreuses qualités. Il se démarque de ses concurrents par sa splendide interface au thème sombre. L'utilisateur se retrouve aisément avec la palette d'outils à sa disposition. Il offre aussi une panoplie de possibilités quant à l'édition des touches : l'utilisateur peut dessiner la forme de son choix ainsi que le nombre de côtés qu'il souhaite, choisir la couleur de plus que l'épaisseur de chaque ligne de sa forme et finalement choisir une image de fond plutôt qu'une couleur. Il peut également modifier l'apparence de la touche quand elle est enfoncée. Au niveau sonore, l'utilisateur a accès à une enveloppe ADSR (*Attack-Decay-Sustain-Release*) comme dans un synthétiseur professionnel, de même qu'une représentation visuelle de son enveloppe. Le mode de lecture automatique est aussi un point fort de notre Gaudrophone. Semblable au jeu *Guitar Hero*, l'aspect visuel est très ludique et très facile d'utilisation, c'est pourquoi on le surnomme le *Jodrofun Hero™*. Pour finir, il est important de mentionner que notre application supporte le matériel MIDI, c'est-à-dire que l'utilisateur peut brancher un contrôleur MIDI (comme un clavier ou un *drum pad*) et lier automatiquement (ou individuellement) son instrument réel à sa création dans le Gaudrophone.

Bien que notre Gaudrophone ait l'air divin, il reste quelques points faibles à l'application. Par exemple, l'interface ne peut pas se rafraîchir lors de la génération d'un nouvel instrument. Cela est dû au *framework* utilisé qui n'actualise pas l'interface malgré la demande. Parfois, l'interface arrive mal à dessiner certaines formes (comme l'hexagone et le dodécagone) et l'affichage vacille ou les bordures ne s'affichent pas comme il le faudrait. Cela est dû au calculs réalisés pour obtenir des coins plus doux et éviter la superposition de bordures. Au niveau de l'audio, la conception du système fait en sorte qu'une même note ne peut se superposer : si on rejoue la note durant la phase *release* du son, celui-ci coupe et on entend le nouveau débuté normalement.

Pour être commercialisé, notre Gaudrophone aurait besoin de quelques ajouts supplémentaire. Il faudrait notamment que les points faibles mentionnés plus-haut soient tous corrigés d'abord. En plus, il faudrait y ajouter le support des fichiers *mp3* comme type de fichier pouvant être associé à une touche, étant donné qu'il s'agit d'un format les plus utilisés. Plutôt que d'utiliser un fichier texte pour encoder les chansons, le logiciel devrait utiliser les fichiers MIDI. Cela permettrait une plus grande souplesse aux utilisateurs afin de s'échanger leurs chansons, ainsi que de permettre d'utiliser d'autres environnements de développement qui utilise le MIDI (*Ableton Live*, *FL Studio*, *MuseScore*, *Finale*, etc.) La synthèse de son devrait aussi être améliorée, afin de pouvoir opérer un synthétiseur similaire à ce qu'il y a généralement dans le domaine. Par exemple, le synthétiseur pourrait avoir trois oscillateurs à forme d'onde, fréquence et volume variable. Finalement, étant donné la diversité des fonctionnalités du logiciel, il faudrait sans aucun doute offrir un support aux utilisateurs, que ce soit par une série de tutoriels vidéo ou un wiki, ainsi que directement dans le logiciel, afin qu'ils puissent être en mesure de maîtriser leur Gaudrophone facilement.