

Baillargeon, Samuel
111183612

Gélinas, Émile
111183587

Mercier Aubin, Alexandre
111180756

Venables, Alexandre
111184382

Génie logiciel orienté objet
GLO-2004, NRC : 89638 (sect. A)

Gaudrophone

Travail présenté à
Jonathan Gaudreault

Faculté de science et génie
Université Laval
A17

Table des matières

1	Énoncé de vision	2
1.1	Introduction	2
1.2	Opportunité d'affaire	2
1.3	Liste des fonctionnalités	2
1.4	Liste des avantages	3
1.5	Autres exigences et contraintes	3
1.6	Capture d'écran	4
2	Diagramme de classe de conception	5
3	Diagramme de séquence de conception	6
3.1	Texte explicatif	7
4	Conclusion	7
4.1	Points forts	7
4.2	Points faibles	8
4.3	Ce qui devrait être amélioré pour la production	8
5	Contribution des membres	8

1 Énoncé de vision

1.1 Introduction

Notre vision de projet est une application qui permet de jouer d'un instrument virtuel. L'instrument peut être créé de toutes pièces par l'utilisateur, en utilisant une interface lui permettant de placer les touches aux endroits qu'il veut et de configurer leur son et leur apparence. L'utilisateur peut également sauvegarder cet instrument pour le réutiliser plus tard. De plus, il peut importer des partitions sous forme de fichiers texte pour faire jouer l'instrument de manière automatique.

1.2 Opportunité d'affaire

De plus en plus de DJ utilisent des panneaux comportant des boutons pour jouer des sons durant leurs concerts. Cet appareil physique est un grand investissement pour quelqu'un qui veut commencer. Le Gaudrophone représente une manière facile et gratuite de se lancer dans cette activité. De plus, puisque l'instrument est complètement personnalisable, cela offre plus de flexibilité qu'une vraie table de DJ. Nous croyons que ce marché sera grandement intéressé par ce produit.

1.3 Liste des fonctionnalités

L'utilisateur doit pouvoir créer et modifier un instrument de tel façon qu'il peut placer des touches ou paramétrer le son et apparence des touches.

Il doit être possible de générer un instrument à l'aide de gabarits préprogrammés et de jouer de son instrument. Lorsque l'utilisateur joue de l'instrument, il peut rejouer de manière automatisée les dernières notes. Si l'utilisateur souhaite un support pour conserver son rythme, il a accès à un métronome.

L'application permet d'enregistrer, de réutiliser les instruments, d'importer une partition pour jouer automatiquement, ainsi qu'apprendre à jouer de cette partition avec le mode muet.

L'utilisateur à accès à un menu d'aide pour apprendre à utiliser l'application avec le menu d'aide.

1.4 Liste des avantages

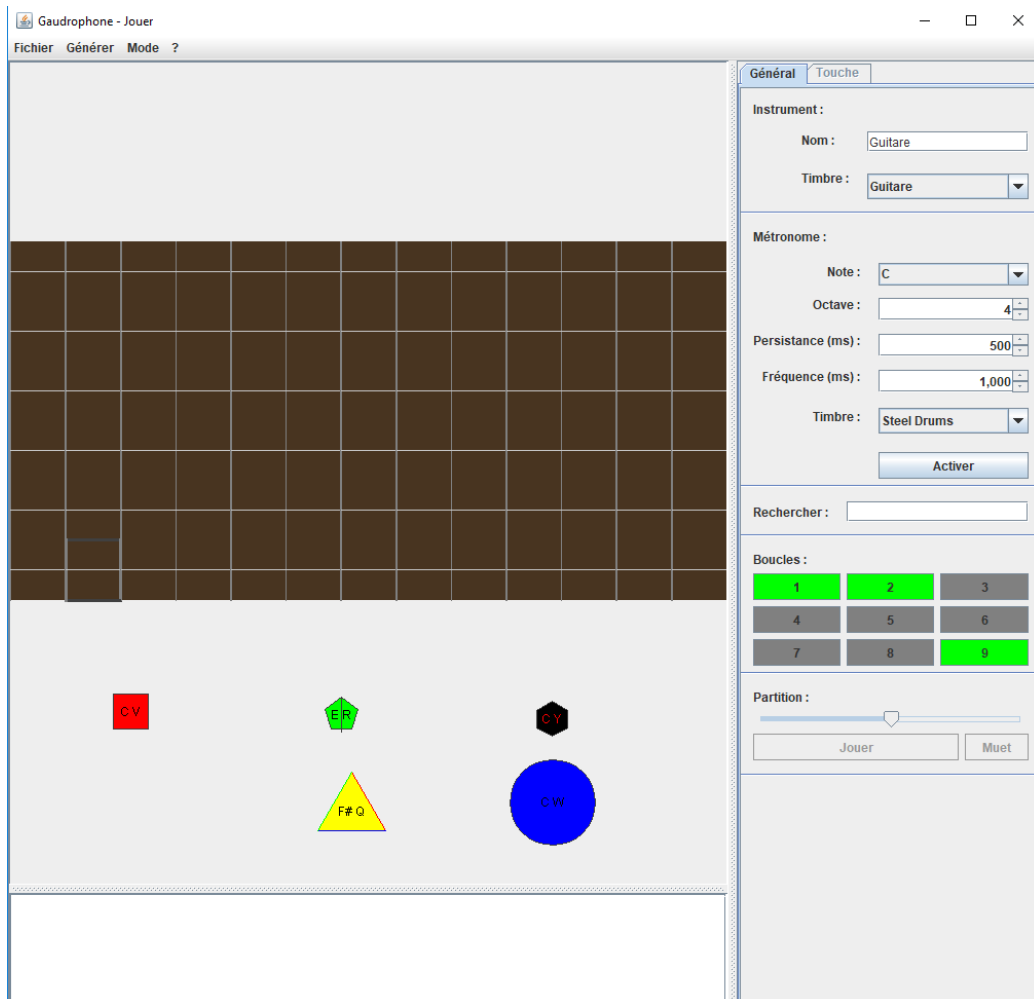
L'application offre des avantages particuliers tels que la possibilité de maîtriser l'art de la musique et en entendant de façon interactive les sons différents et variés de l'industrie musicale. L'instrument permet la personnalisation d'un instrument au goût de l'utilisateur et de simuler l'équivalent d'un groupe complet de façon numérique. L'interface adaptative sera simple d'utilisation pour tous. L'application sera portable, soit compatible avec écran tactile.

Comparé à un ensemble complet de mixage, le programme est offert à un prix compétitif.

1.5 Autres exigences et contraintes

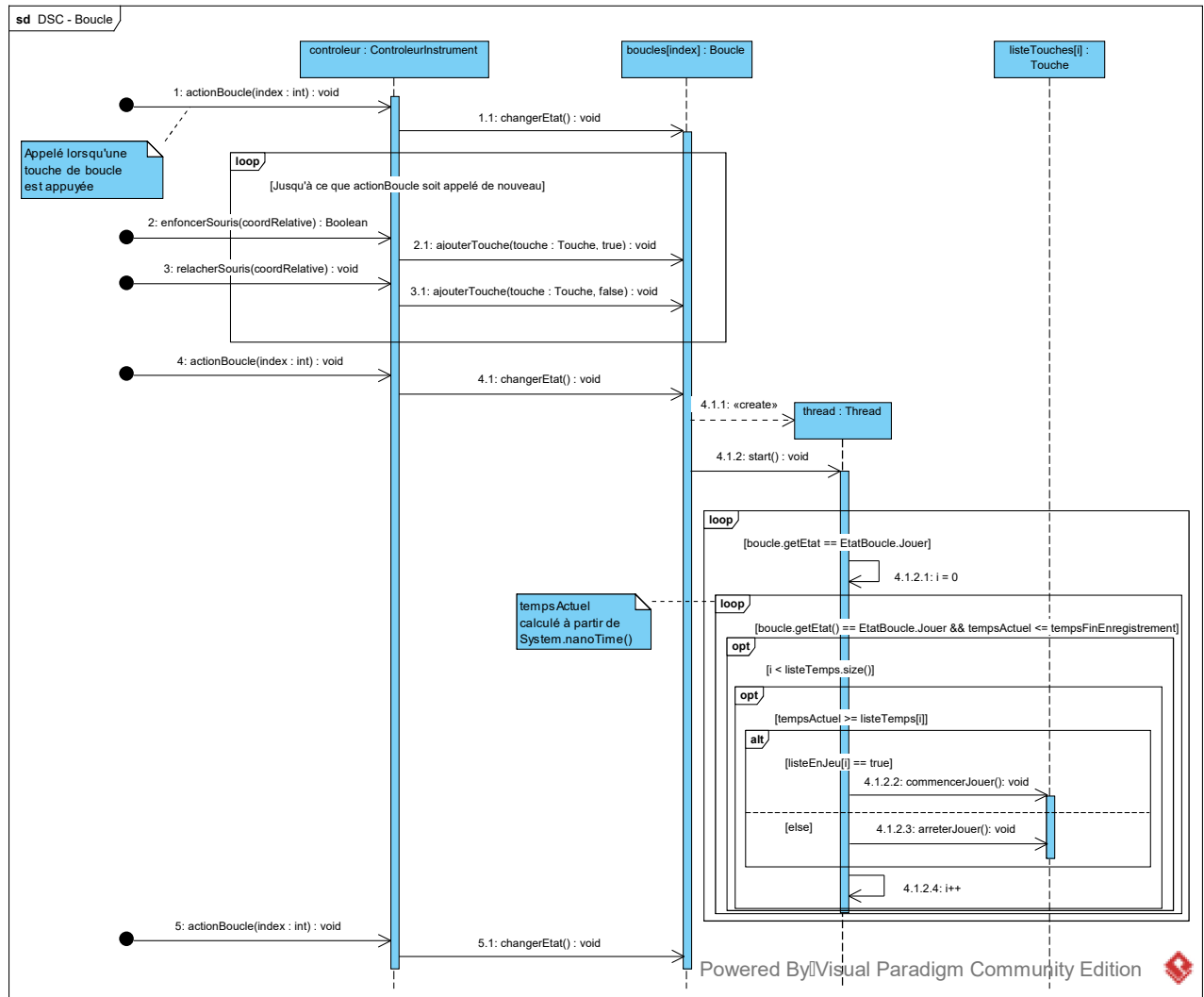
L'application devra aussi se soumettre à d'autres exigences. Il devra produire un son agréable à l'oreille, ne pas représenter de délais entre le moment où l'utilisateur appuie sur la touche et le moment où le son est produit. Les boîtes de dialogues flottantes ne seront pas permises et la fenêtre devra être redimensionnable, tout en conservant une taille proportionnelle des boutons et dessins.

1.6 Capture d'écran





3 Diagramme de séquence de conception



3.1 Texte explicatif

Ce texte prend en compte que la pédale de bouclage utilisée est la pédale numéro 1. Au départ, la boucle est à l'état "Inactif". Par la suite, lorsque l'utilisateur appuie sur la touche "1" du clavier ou sur le bouton de la boucle 1, le contrôleur reçoit l'appel `actionBoucle(0)`. Il indique ensuite à la boucle 1 de changer son état. La boucle se met donc en état "Enregistrer".

Une fois dans ce mode, les différentes touches appuyées sont enregistrées lorsqu'elles sont appuyées avec le contrôleur qui appelle `ajouterTouche()` sur la boucle. Lorsque l'utilisateur appuie une seconde fois sur la pédale 1, la boucle change d'état de la même façon que plus tôt pour se mettre dans le mode "Jouer".

Lorsque la boucle arrive dans ce mode, elle active aussitôt une thread qui servira à jouer les différentes touches qui ont été enregistrées dans la liste durant l'état "Enregistrer". Dans cette thread, on commence par obtenir le temps de départ avec `System.nanoTime()`, puis, on commence une boucle qui va continuellement aller vérifier `System.nanoTime()` et le comparer au temps de départ. Lorsqu'on arrive au prochain temps enregistré dans la liste, on joue la note correspondante, puis on continue à attendre d'arriver au temps du prochain index. Lorsqu'on atteint le temps de fin de la boucle, on redémarre du départ avec l'index 0 et un nouveau temps de départ.

Cette boucle continue jusqu'à ce que l'utilisateur appuie une troisième fois sur la pédale de bouclage. À ce moment, l'état de la boucle est remis à "Inactif" et la thread s'arrête d'elle-même. La boucle est maintenant prête à enregistrer une nouvelle séquence en recommençant du début du diagramme.

4 Conclusion

4.1 Points forts

Lors de la conception de l'application, nous avons réussi à incorporer une certaine convivialité dans l'utilisation de l'interface. Par exemple, lors de l'ajout ou du déplacement d'une touche, il est possible de voir la touche être redessinée à l'emplacement de la souris. Les points demandés par l'utilisateur

initialement sont intégrés en respectant les critères initiaux.

De plus, nous avons intégré une fonctionnalité supplémentaire à ce qui était demandé: on peut associer une touche du clavier à chaque touche de l'instrument pour pouvoir jouer de l'instrument sans se servir de la souris et même jouer plusieurs notes en même temps.

4.2 Points faibles

Lors de la redimension de l'interface, un carré noir apparaît avant que l'application ne redessine son contenu. Ce phénomène est sans doute causé par l'interface java swing qui n'est pas adaptée pour des productions de masses. Il existe encore des bugs dans notre application devant être résolus, ce qui est normal considérant que nous ne possédons pas d'équipe de testeurs.

4.3 Ce qui devrait être amélioré pour la production

Pour la production, une refonte de l'interface devrait être faite afin d'inclure plus d'interactivité ainsi qu'une présentation plus attirante pour l'oeil. Il serait aussi pertinent de choisir à partir d'une plus grande diversité de timbres et de générateurs d'instruments. Une détection des collisions plus précise devrait être appliquée. Toutefois, nous sommes convaincu qu'avec quelques retouches, il serait possible d'offrir une application digne de production.

5 Contribution des membres

1. Métronome: Alexandre Mercier-Aubin
2. Bouclage: Samuel Baillargeon
3. Gestion du clavier: Samuel Baillargeon
4. Gestion des partitions: Alexandre Venables
5. Rapport du Livrable 4: Émile Gélinas et Alexandre Mercier-Aubin
6. Mise à jour du diagramme de classes : Émile Gélinas
7. Modification dans l'interface : Émile Gélinas

8. Diagramme de séquences du bouclage : Samuel Baillargeon, Émile Gélinas et Alexandre Mercier-Aubin