

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

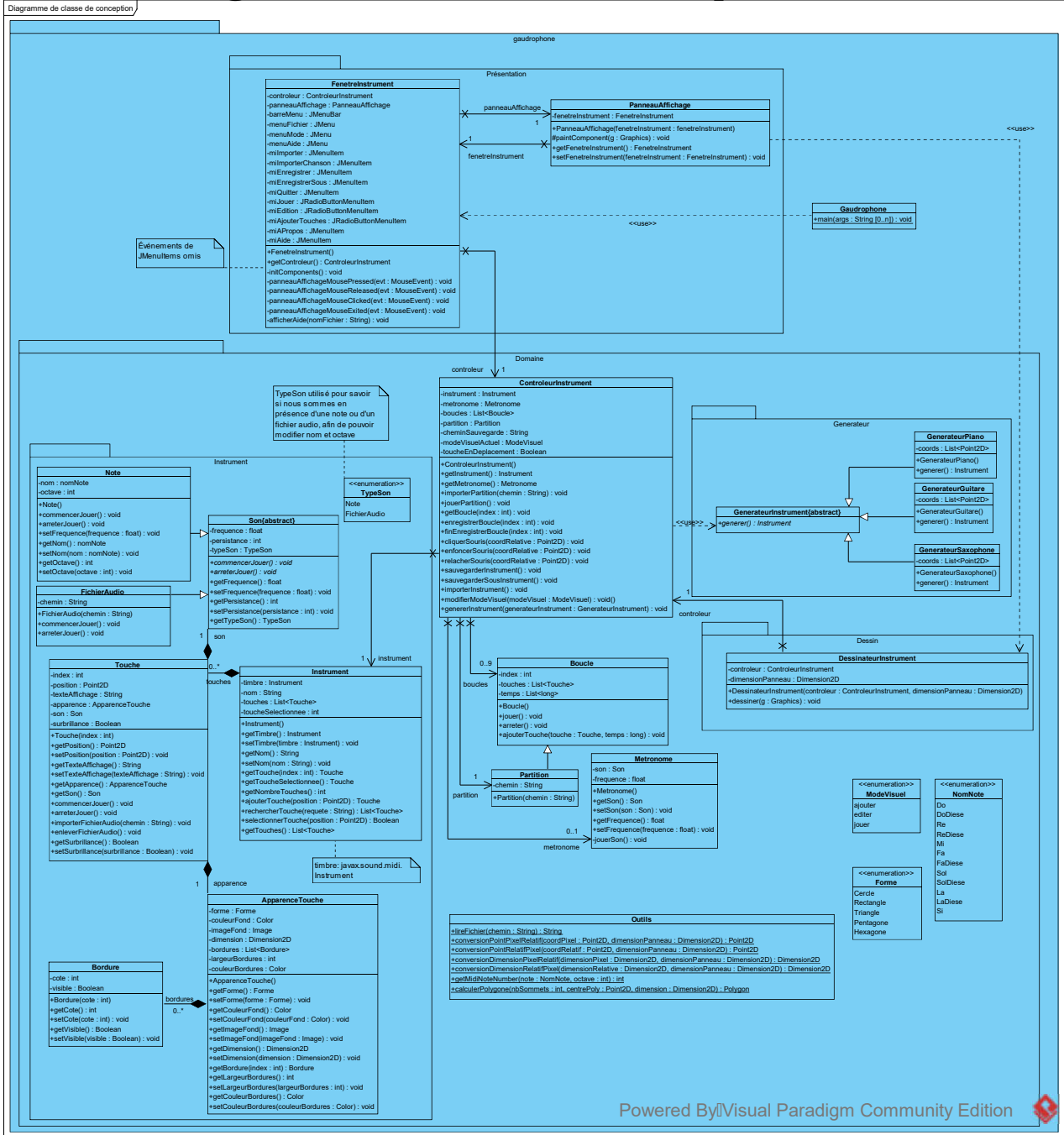
Faculté de science et génie
Université Laval
A17

Table des matières

1	Diagramme de classe de conception	3
1.1	Description des classes	4
2	Diagramme de package	8
2.1	Texte explicatif	9
3	Diagrammes de séquence de conception	10
3.1	Conversion des coordonnées	10
3.1.1	Description	10
3.2	Clic sur touche	10
3.2.1	Description	10
3.3	Ajout d'une touche	11
3.3.1	Description	11
3.4	Déplacement d'une touche	12
3.4.1	Description	12
3.5	Clic sur une touche pour produire le son	13
3.5.1	Description	14
3.6	Affichage de l'instrument	15
3.6.1	Description	16
3.7	Génération de la guitare	17
3.7.1	Description	18
4	Diagramme de Gantt	19
4.1	Texte explicatif	20
5	Contribution des membres	20
6	Annexe	21
6.1	Énoncé vision	21
6.1.1	Introduction	21
6.1.2	Opportunité d'affaire	21
6.1.3	Liste des fonctionnalités	21
6.1.4	Liste des avantages	22
6.1.5	Autres exigences et contraintes	22
6.2	Modèle du domaine	23
6.3	Modèle des cas d'utilisation	24
6.4	Glossaire	25

6.5	Note Midi	25
7	Bibliographie	26

1 Diagramme de classe de conception



1.1 Description des classes

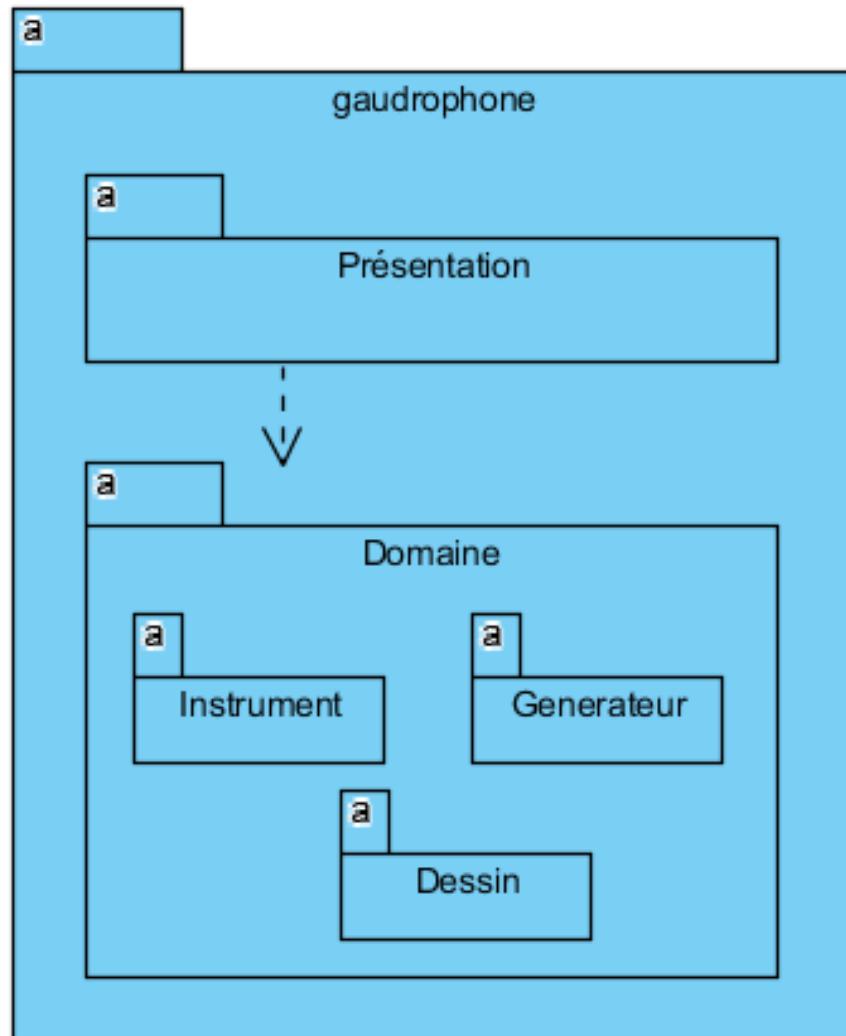
Classe	Description
FenetreInstrument	La fenêtre principale de l'application permettant à l'utilisateur de contrôler les différentes fonctionnalités du programme de manière conviviale. Elle accède au reste des classes du domaine en passant par le <u>ControleurInstrument</u> . La fenêtre comporte également le <u>PanneauAffichage</u> .
PanneauAffichage	Le panneau d'affichage contenu dans la <u>fenetreInstrument</u> permet d'afficher graphiquement l' <u>instrument</u> . Il utilise le <u>dessinateurInstrument</u> afin d'effectuer le dessin des éléments de l' <u>instrument</u> .
Gaudrophone	Point d'entrée de l'application. Cette classe sert seulement à contenir la fonction <code>main()</code> qui instancie la <u>FenetreInstrument</u> et qui l'affiche.
ModeVisuel «enumeration»	Énumération servant à représenter le mode visuel actuel de l'application, pouvant être : jouer de l'instrument, éditer l'instrument ou ajouter des touches.
ControleurInstrument	Contrôleur de Larman servant d'intermédiaire entre la couche Présentation et la couche Domaine de l'application. C'est également lui qui conserve les informations globales telles que le mode actuel de l'application, ainsi que les instances des classes <u>Instrument</u> , <u>Metronome</u> , <u>Partition</u> , ainsi que les <u>Boucles</u> .
Instrument	Contient principalement la liste des <u>touches</u> et leur gestion, mais aussi des informations plus globales comme le nom et le timbre de l'instrument.

Touche	Une touche est contenue dans un instrument. Elle possède un son, qui peut être une <u>note</u> ou un <u>fichier audio</u> . Elle possède aussi une instance d' <u>apparence</u> lui offrant les caractéristiques visuelles qui seront affichées par le <u>dessinateurInstrument</u> .
ApparenceTouche	L'apparence de la touche regroupe les caractéristiques visuelles d'une <u>touche</u> . Ces caractéristiques seront par la suite utilisées par le <u>dessinateurInstrument</u> afin d'afficher les touches. Une apparence contient une liste de <u>bordures</u> qui représentent l'apparence de chacune des lignes entourant la <u>touche</u> .
Forme «enumeration»	Énumération représentant la forme d'une <u>Touche</u> , pouvant être : cercle, triangle, rectangle, pentagone ou hexagone.
Bordure	La bordure est l'un des traits entourant une <u>touche</u> . Elle peut être visible ou non. Le nombre de bordure pour une touche est prédéterminé selon sa forme.
<i>Son</i>	Son est une classe abstraite représentant les multiples bruits qu'une <u>touche</u> peut faire. Les sons peuvent être soit une <u>note</u> ou un <u>fichier audio</u> .
TypeSon «enumeration»	Énumération servant à représenter le type de son qu'a la classe <u>son</u> , pouvant être : note ou fichier audio
Note	Une note hérite de <u>son</u> , elle possède un nom de note et un octave en plus des éléments de son. Elle construit par dessus la fonction set-Frequency pour modifier accordément l'octave et la note si la fréquence est modifiée.
NomNote «enumeration»	Énumération servant à représenter toutes les noms de notes possible dans une <u>Note</u> , pouvant être Do, Do♯, Ré, Ré♯, Mi, Fa, Fa♯, Sol, Sol♯, La, La♯ ou Si

Boucle	La boucle représente une série de <u>touches</u> conservées dans un ordre et selon un temps. Lors de l'enregistrement d'une boucle, le <u>ControleurInstrument</u> instancie un chronomètre et le démarre. Jusqu'à ce que <u>finEnregistrer()</u> soit appelé dans le contrôleur, chaque touche pressée est ajoutée dans la boucle ainsi que son temps. Une fois terminé, la boucle permet de jouer la série de touches enregistrées à répétition.
Partition	Une partition est un objet accédant à un fichier binaire permettant d'importer une série de touches pressées ainsi que le temps auquel elles ont été pressées. Vu sa similarité avec une <u>boucle</u> , elle implémentera la même structure.
Metronome	Le métronome est une classe contenant un <u>son</u> et une fréquence. Lorsque activé, un chronomètre récursif émettra le son à chaque fin de fréquence. Lorsque désactivé, le chronomètre s'arrêtera, finissant ainsi le cycle récursif. Le métronome permettra ainsi de faciliter l'enregistrement ou la lecture des <u>partitions</u> pour l'utilisateur en lui fournissant un rythme.
<i>GenerateurInstrument</i>	Le générateur d'instrument est une classe abstraite représentant le patron de la génération automatisée d' <u>instrument</u> .
GenerateurPiano	Le générateur de piano hérite de la classe abstraite <u>GenerateurInstrument</u> . Elle sert à générer un <u>instrument</u> ayant l'apparence et le son d'un piano.
GenerateurGuitare	Le générateur de guitare hérite de la classe abstraite <u>GenerateurInstrument</u> . Elle sert à générer un <u>instrument</u> ayant l'apparence et le son d'une guitare.
GenerateurSaxophone	Le générateur de saxophone hérite de la classe abstraite <u>GenerateurInstrument</u> . Elle sert à générer un <u>instrument</u> ayant l'apparence et le son d'un saxophone.

FichierAudio	Un fichier audio hérite de <u>son</u> , il contient le chemin vers un fichier audio au choix de l'utilisateur qui pourra être lu afin de produire un <u>son</u> .
DessinateurInstrument	Permet d'afficher la représentation graphique de l' <u>instrument</u> sur le <u>PanneauAffichage</u> . Il utilise le <u>contrôleur</u> pour avoir accès aux informations de plusieurs sources stratégiques telles que l' <u>apparence</u> de chaque <u>touche</u> .
Outils	La classe outils contient uniquement des fonctions statiques pouvant être aisément réutilisable dans le projet ainsi que d'éventuels autres projets. Entre autre, la lecture de fichier binaire vers une variable string peut être utilisée dans l'interface pour le menu d'aide et potentiellement dans la lecture des partitions.

2 Diagramme de package



2.1 Texte explicatif

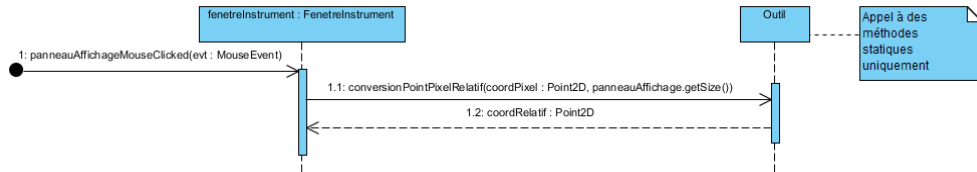
Notre application utilise 6 packages:

Package	Description
gaudrophone	Package principal englobant tous les autres packages. La seule classe faisant directement partie de ce package est <u>Gaudrophone</u> , le point d'entrée de l'application.
Présentation	Sous-package de <u>gaudrophone</u> contenant les classes en lien avec l'affichage graphique de l'application. Peut communiquer avec <u>Domaine</u> .
Domaine	Sous-package de <u>gaudrophone</u> contenant toutes les classes servant à la logique applicative. Ne peut pas communiquer avec <u>Présentation</u> .
Instrument	Sous-package de <u>Domaine</u> contenant la classe <u>Instrument</u> , ainsi que toutes les autres classes qui représentent des parties de l' <u>Instrument</u> .
Generateur	Sous-package de <u>Domaine</u> contenant la classe abstraite <u>GenerateurInstrument</u> ainsi que toutes les classes en héritant.
Dessin	Sous-package de <u>Domaine</u> contenant la classe <u>DessinateurInstrument</u> , qui sert à dessiner une représentation graphique de l' <u>Instrument</u> dans la fenêtre de l'application.

Pour voir quelles classes font partie de quel package, référez-vous au diagramme de classe.

3 Diagrammes de séquence de conception

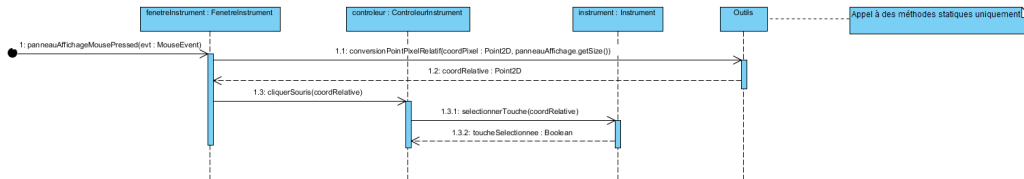
3.1 Conversion des coordonnées



3.1.1 Description

Lorsque l'utilisateur effectue un clic dans le panneau d'affichage, la fenetre-Instrument reçoit l'événement `panneauAffichageMouseClicked()`. Dans cet événement, on appelle la méthode statique `Outils.conversionPixelRelatif()` pour convertir les coordonnées pixel reçues dans l'événement en coordonnées relatives.

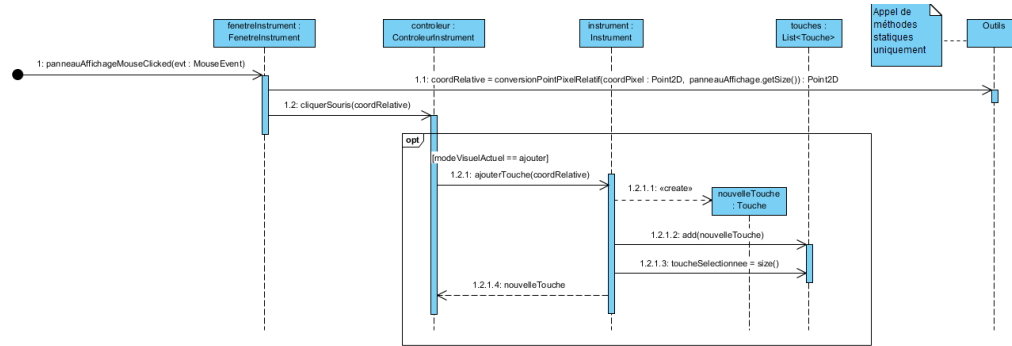
3.2 Clic sur touche



3.2.1 Description

Lorsque l'utilisateur effectue un clic dans le panneau d'affichage, la fenetre-Instrument reçoit l'événement `panneauAffichageMouseClicked()`. Dans cet événement, on appelle la méthode statique `Outils.conversionPixelRelatif()` pour convertir les coordonnées pixel reçues dans l'événement en coordonnées relatives. Par la suite, on passe cette information au contrôleur, qui appelle à son tour `instrument.selectionnerTouche()` en spécifiant l'emplacement relatif où le bouton de la souris a été cliqué. L'instrument regarde dans la liste des touches. S'il trouve une touche à l'emplacement spécifié, il met l'index de cette touche en tant que `toucheSelectionnee` et il retourne `true` au contrôleur, qui lève son flag `toucheSelectionnee`.

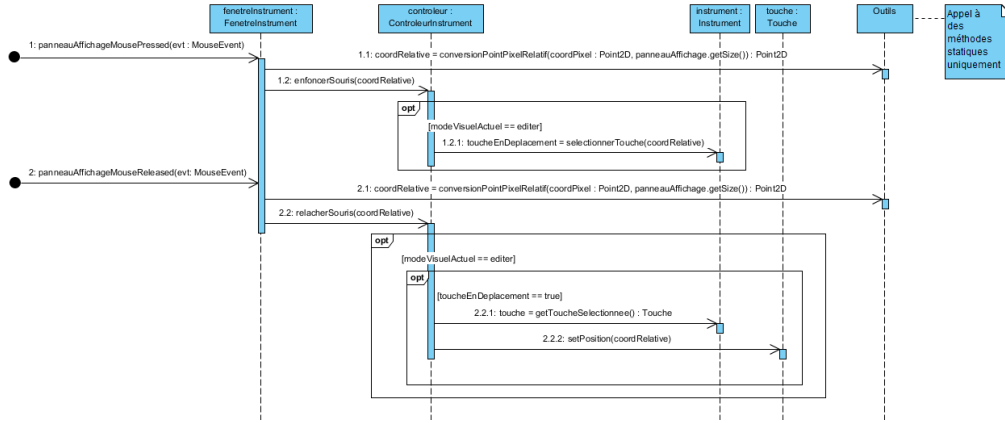
3.3 Ajout d'une touche



3.3.1 Description

Lorsque l'application est en mode ajouter (préalablement sélectionné par l'utilisateur dans le menu Mode), et que l'utilisateur clique à un emplacement quelconque dans le panneau d'affichage, la fenetreInstrument reçoit l'événement `panneauAffichageMouseClicked()`. Dans cet événement, on appelle la méthode statique `Outils.conversionPixelRelatif()` pour convertir les coordonnées pixel reçues dans l'événement en coordonnées relatives. Ensuite, on passe cette information au contrôleur, qui appelle à son tour `instrument.ajouterTouche()` en spécifiant l'emplacement relatif qui a été cliqué. L'instrument crée donc une nouvelle instance de la classe `Touche`, l'ajoute à la liste, fait en sorte que cette nouvelle touche soit celle qui est sélectionnée, et la retourne au contrôleur.

3.4 Déplacement d'une touche

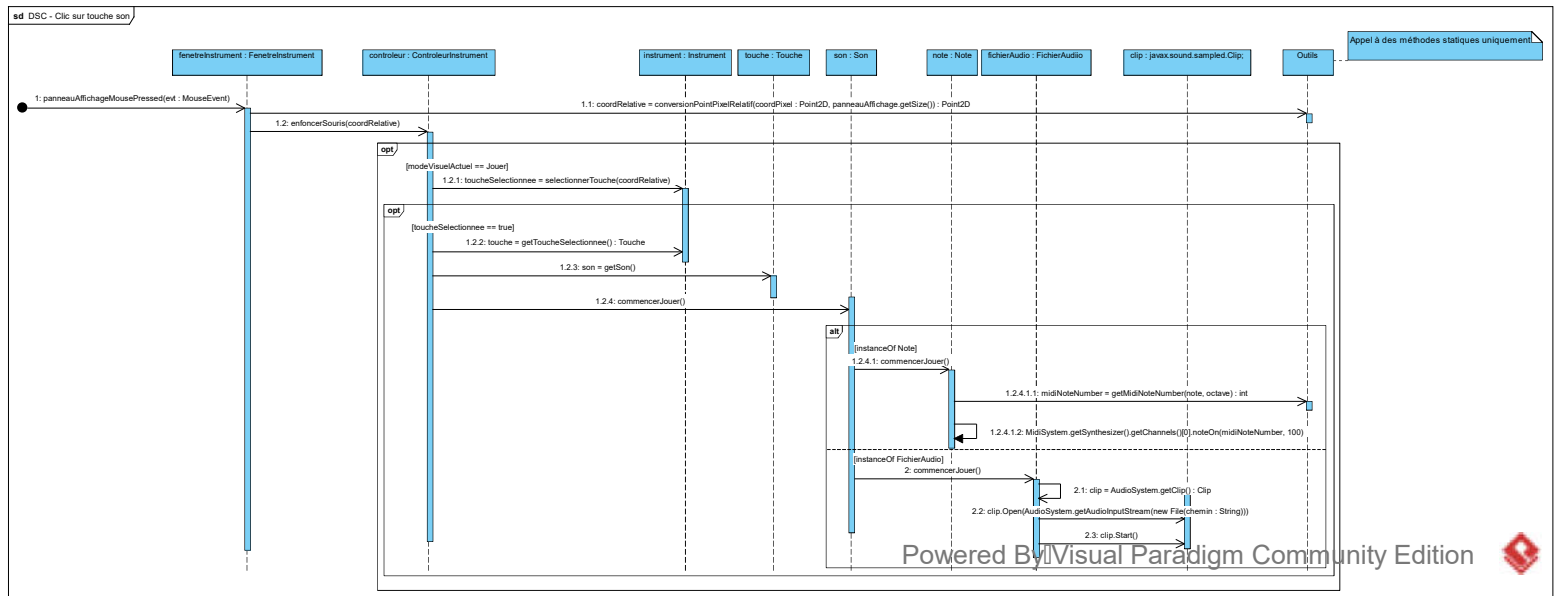


3.4.1 Description

Lorsque l'application est en mode édition (préalablement sélectionné par l'utilisateur dans le menu Mode), et que l'utilisateur enfonce le bouton de sa souris en pointant une touche existante, la fenetreInstrument reçoit l'événement `panneauAffichageMousePressed()`. Dans cet événement, on appelle la méthode statique `Outils.conversionPixelRelatif()` pour convertir les coordonnées pixel reçues dans l'événement en coordonnées relatives. Ensuite, on passe cette information au contrôleur, qui appelle à son tour `instrument.selectionnerTouche()` en spécifiant l'emplacement relatif où le bouton de la souris a été enfoncé. L'instrument regarde dans la liste des touches. S'il trouve une touche à l'emplacement spécifié, il met l'index de cette touche en tant que `toucheSelectionnee` et il retourne `true` au contrôleur, qui lève son flag `toucheEnDeplacement` pour se rappeler qu'une touche est en train d'être déplacée.

Par la suite, lorsque l'utilisateur relâche le bouton de la souris (après l'avoir déplacée), la fenetreInstrument reçoit l'événement `panneauAffichageMouseReleased()`. Dans cet événement, on utilise une fois encore `Outils.conversionPixelRelatif()` pour convertir les coordonnées pixel en coordonnées relatives, qu'on passe une fois de plus au contrôleur. Ensuite, le contrôleur vérifie si le flag `toucheEnDeplacement` est levé. Si oui, il utilise l'instrument pour trouver la touche qui était sélectionnée, puis il modifie la position de cette dernière pour mettre l'emplacement où l'utilisateur a relâché le bouton de la souris.

3.5 Clic sur une touche pour produire le son

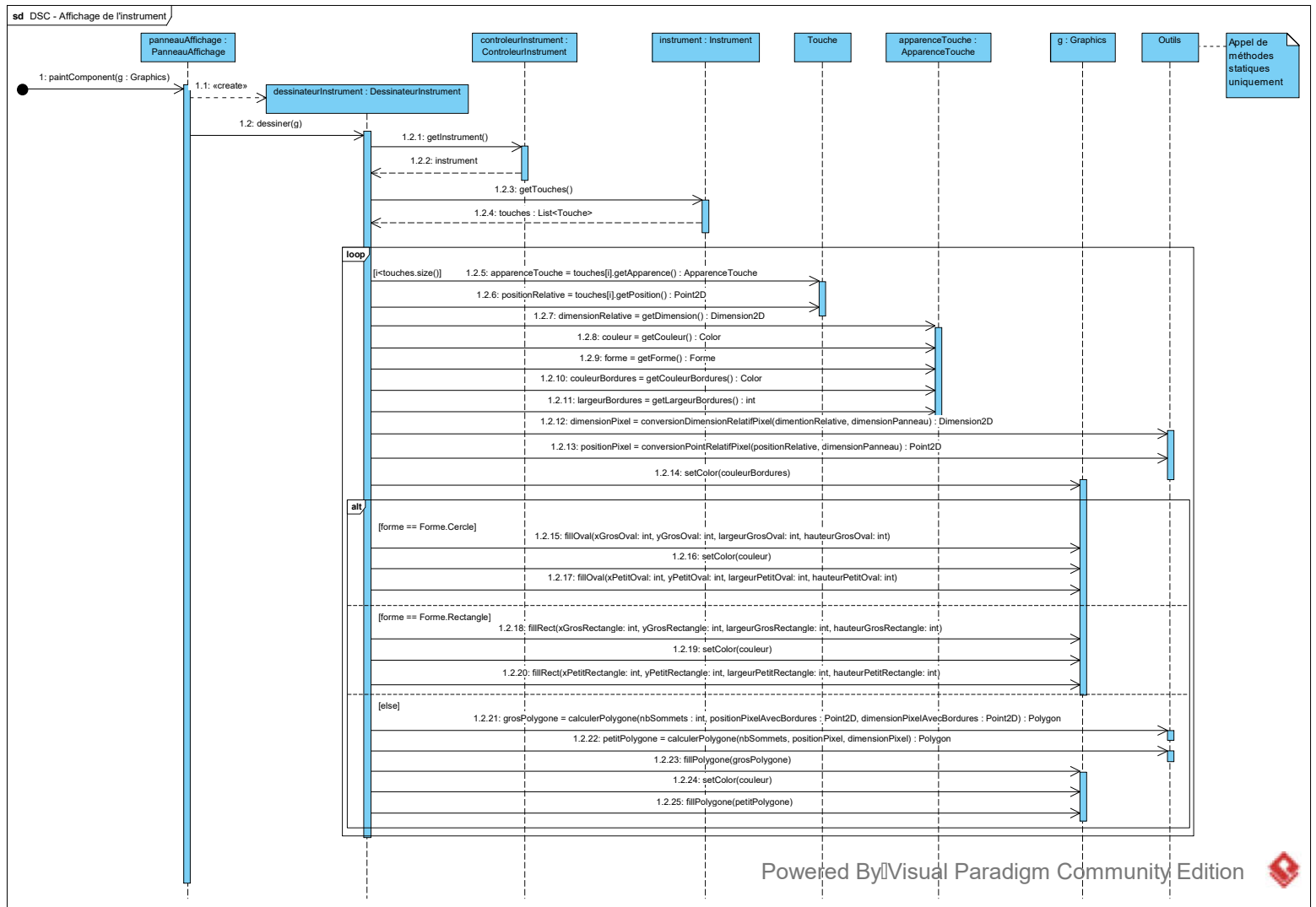


3.5.1 Description

Lorsque l'application est en mode jouer (préalablement sélectionné par l'utilisateur dans le menu Mode), et que l'utilisateur enfonce le bouton de sa souris en pointant une touche existante, la fenetreInstrument reçoit l'événement panneauAffichageMousePressed(). Dans cet événement, on appelle la méthode statique Outils.conversionPixelRelatif() pour convertir les coordonnées pixel reçues dans l'événement en coordonnées relatives. Ensuite, on passe cette information au contrôleur, qui appelle à son tour instrument.sélectionnerTouche() en spécifiant l'emplacement relatif où le bouton de la souris a été enfoncé. L'instrument regarde dans la liste des touches. S'il trouve une touche à l'emplacement spécifié, il met l'index de cette touche en tant que toucheSelectionnee et il retourne true au contrôleur, qui lève son flag toucheSelectionnee pour se rappeler qu'une touche est à été cliquer pour émettre un son.

Ensuite, le contrôleur va chercher le son associé à cette touche, puis il appelle la méthode commencerJouer(). La méthode peut être définie de 2 façons différentes, tout dépendant si le son est une note ou bien un fichier audio. Si le son est une note : la méthode commencerJouer() fait appel à la méthode statique Outils.getMidiNoteNumber() pour convertir la note et l'octave en une note Midi (voir en annexe 6.6 le tableau des notes Midi). Par la suite, on fait appel à la méthode MidiSystem.getSynthesizer().getChannels()[0].noteOn() qui prend en paramètre la note Midi et le volume, pour ensuite jouer le son de la note. Si le son est un fichier audio : la méthode commencerJouer() crée une instance de Clip à l'aide de AudioSystem.getClip(). Par la suite, on configure le fichier audio dans le clip à l'aide de clip.Open(AudioSystem.getAudioInputStream(new File())) où le paramètre est le chemin du fichier audio. Pour émettre le son, on utilise clip.Start() qui commence la lecture du fichier audio.

3.6 Affichage de l'instrument

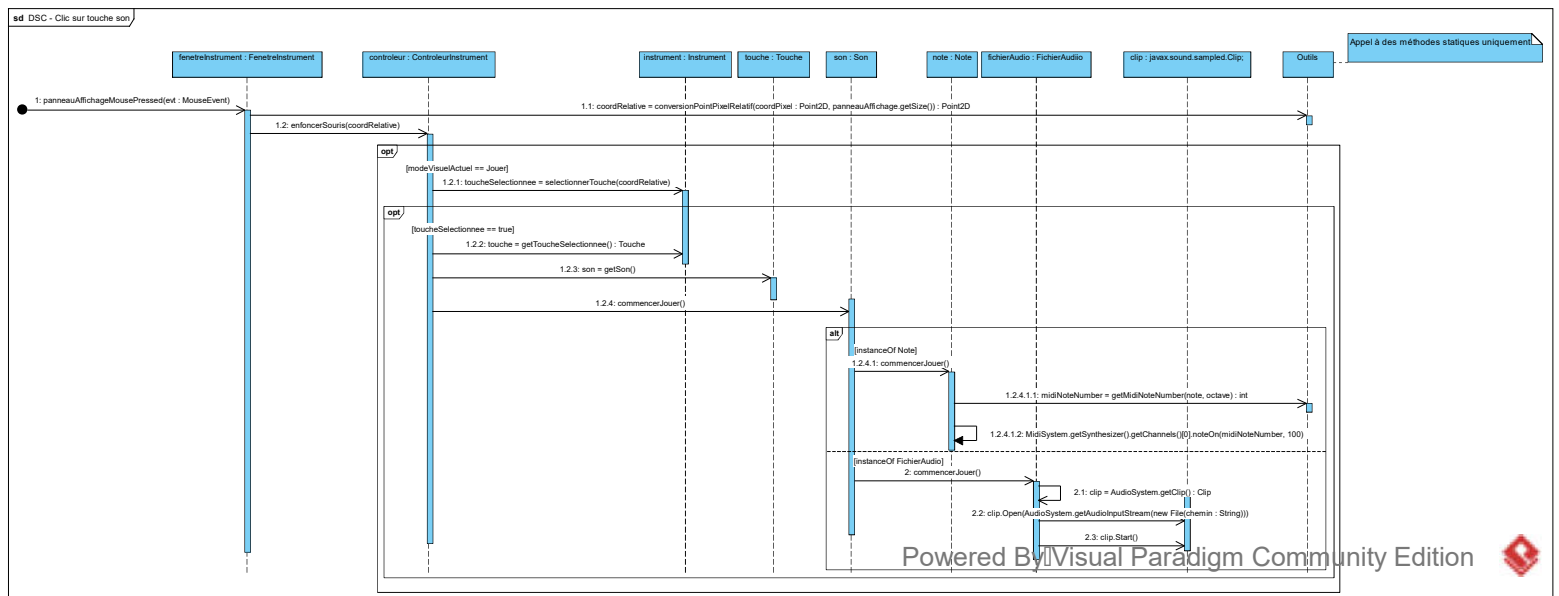


3.6.1 Description

Lorsque le panneau d’affichage reçoit `paintComponent()` en événement, il crée une instance de `DessinateurInstrument` et appelle sa fonction `dessiner()` en lui donnant en paramètre un objet `Graphics`. Par la suite, le dessinateurInstrument va chercher l’instrument et la liste des touches reliées à l’instrument. À l’intérieur d’une boucle qui passe chacune des touches dans la liste, on va chercher son apparence et sa position relative. Dans son apparence, on va chercher sa dimension relative, sa forme, sa couleur, la couleur de ses bordures et la largeur de ses bordures. Par la suite, on va calculer sa dimension et sa position en pixel à l’aide de la classe outils. On attribut la couleur des bordures à l’objet `Graphics`.

Nous avons ensuite trois cas possible : on dessine soit un cercle, un rectangle ou un polygone. Pour le cas des cercles, on dessine un gros cercle dont les coordonnées sont calculées en additionnant la largeur des bordures aux informations recueillies sur la touche. Ce gros cercle sert à faire la bordure de notre touche. Par la suite, on change la couleur de notre objet `Graphics` pour la couleur de la touche et on redessine un cercle plus petit à l’intérieur de l’autre cercle. Pour le cas des rectangles, c’est le même principe que les cercles, mais on dessine des rectangles à la place. Pour finir, le cas des polygones. Le nombre de sommets est défini avec la forme (par exemple, un triangle aurait 3 sommets). On calcule par la suite deux polygones, un gros et un petit à l’aide de la fonction `calculerPolygone` de notre objet Outils. On dessine un gros polygone avec les informations recueillies pour faire les bordures de notre touche. On change la couleur de l’objet `Graphics` pour la couleur de notre touche et on dessine le petit polygone.

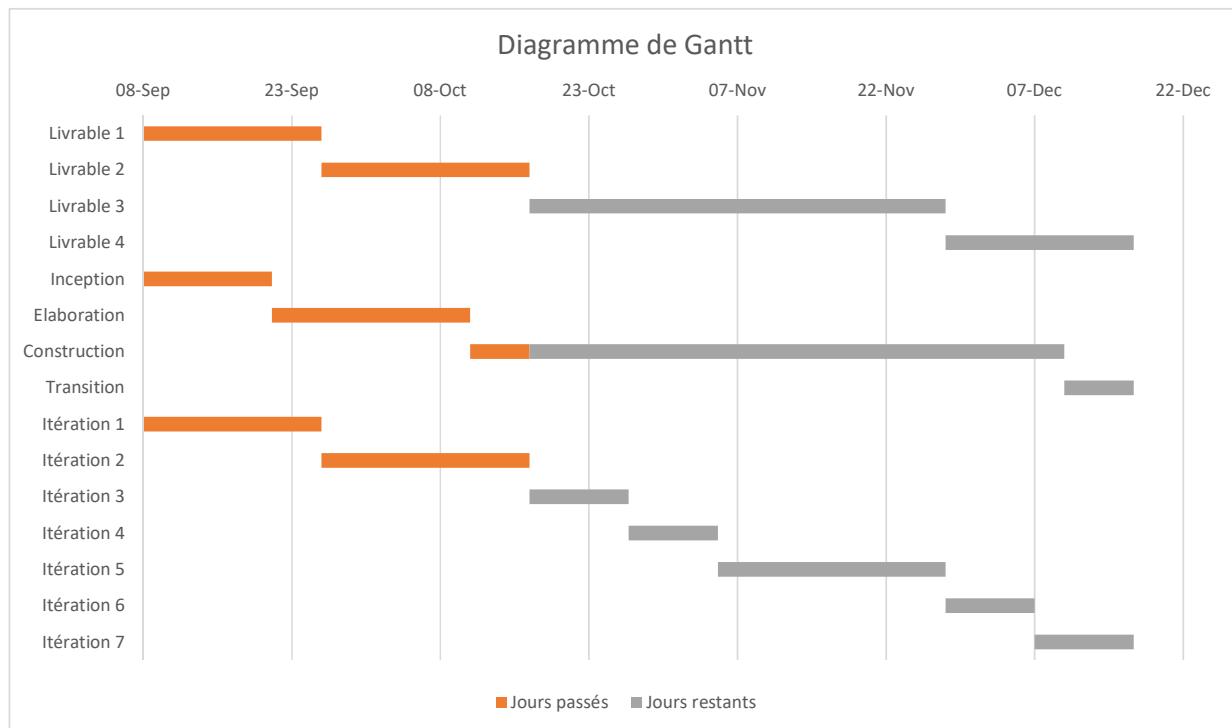
3.7 Génération de la guitare



3.7.1 Description

Lorsque le contrôleur reçoit un appel pour générer une guitare, il appelle la fonction `generer()` de la classe `GenerateurGuitare`, qui retourne un instrument. Par la suite, `GenerateurGuitare` crée une nouvelle instance d'un instrument et y associe le timbre d'une guitare. Dans une boucle qui passe tous les éléments d'une liste de coordonnées, le `GenerateurGuitare` appelle la fonction `ajouterTouche()` qui crée une touche à l'emplacement désiré. Son apparence, son son et ses bordures sont aussi créés. Par la suite, on va chercher l'apparence de la touche précédemment créée. Ensuite, on va attribuer une forme rectangulaire et une couleur prédéfinie à l'apparence de notre touche. Dans une boucle qui passe chacune des bordures, on va chercher notre bordure, on la met visible, on lui donne une couleur prédéfinie et une largeur par défaut. Par la suite, on va chercher le son de la touche. On donne une fréquence et une persistance prédéfinie au son. On ajoute la touche dans la liste de touches, on retourne la note au `DessinateurInstrument` et il retourne le nouvelInstrument au contrôleur.

4 Diagramme de Gantt



4.1 Texte explicatif

Itération	Description
Itération 1	Conception de tous les cas d'utilisation.
Itération 2	Modification de la conception des cas d'utilisation selon les commentaires de l'enseignant.
Itération 3	Création des classes.
Itération 4	Implémentation des cas d'utilisation : Nouvel instrument, modifier l'instrument, ajouter une touche, jouer d'un instrument, importer un instrument et sauvegarder un instrument. (Semaine de lecture)
Itération 5	Implémentation des cas d'utilisation : Importer partition, jouer une partition et générer un instrument.
Itération 6	Implémentation des cas d'utilisation : Activer le métronome et faire boucler une séquence.
Itération 7	Vérifications et retouches.

5 Contribution des membres

1. Diagramme de classe de conception : Tous, en groupe
2. Description des classe du diagramme de classe de conception : Alexandre Mercier-Aubin
3. Diagramme de package : Tous, en groupe
4. Description du diagramme de package : Samuel Baillargeon
5. Diagramme de Gantt et sa description : Tous, en groupe
6. Diagrammes de séquence et leur description : Tous, individuellement

6 Annexe

6.1 Énoncé vision

6.1.1 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 en tant que fichier binaire 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.

6.1.2 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.

6.1.3 Liste des fonctionnalités

L'utilisateur doit pouvoir créer et modifier un instrument de telle 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 que d'apprendre à jouer de cette partition avec le mode muet.

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

6.1.4 Liste des avantages

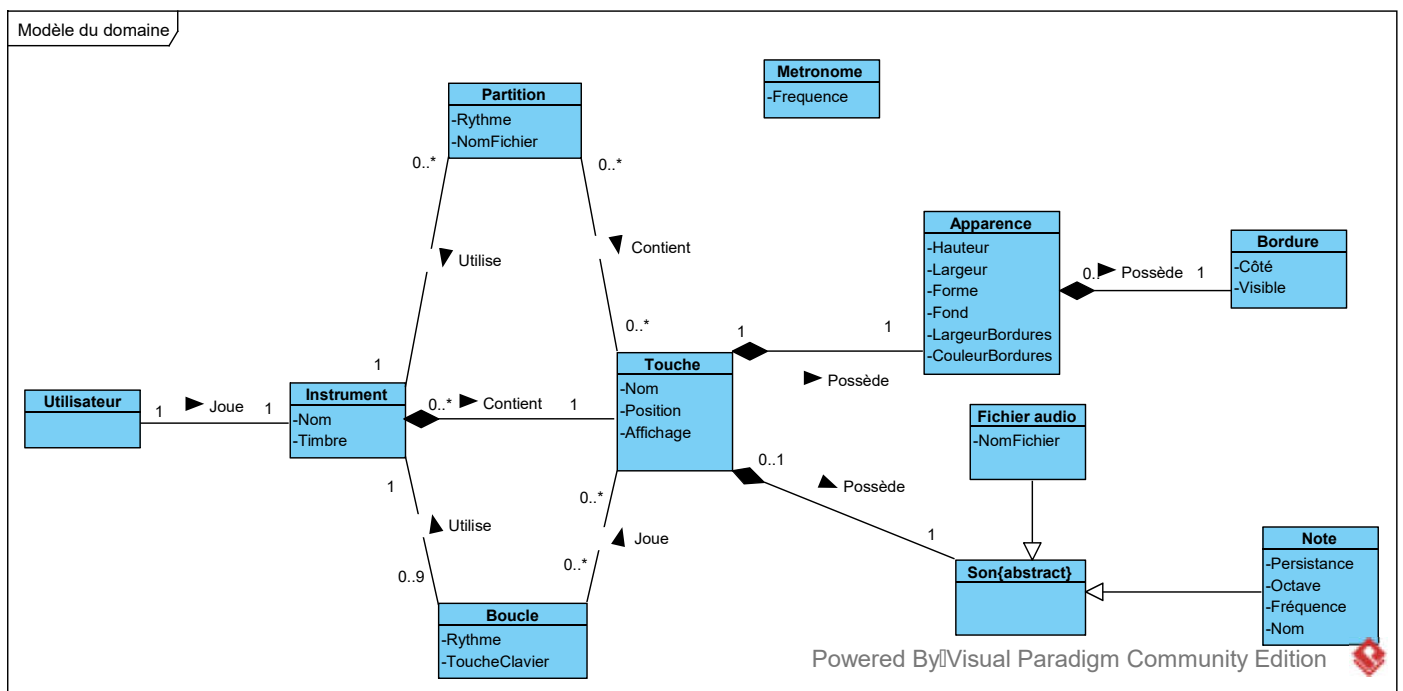
L'application offre des avantages particuliers tels que la possibilité de maîtriser l'art de la musique et en entendant de façon interactive les sons différents et variés de l'industrie musicale. L'instrument permet la personnalisation d'un instrument au goût de l'utilisateur et de simuler l'équivalent 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.

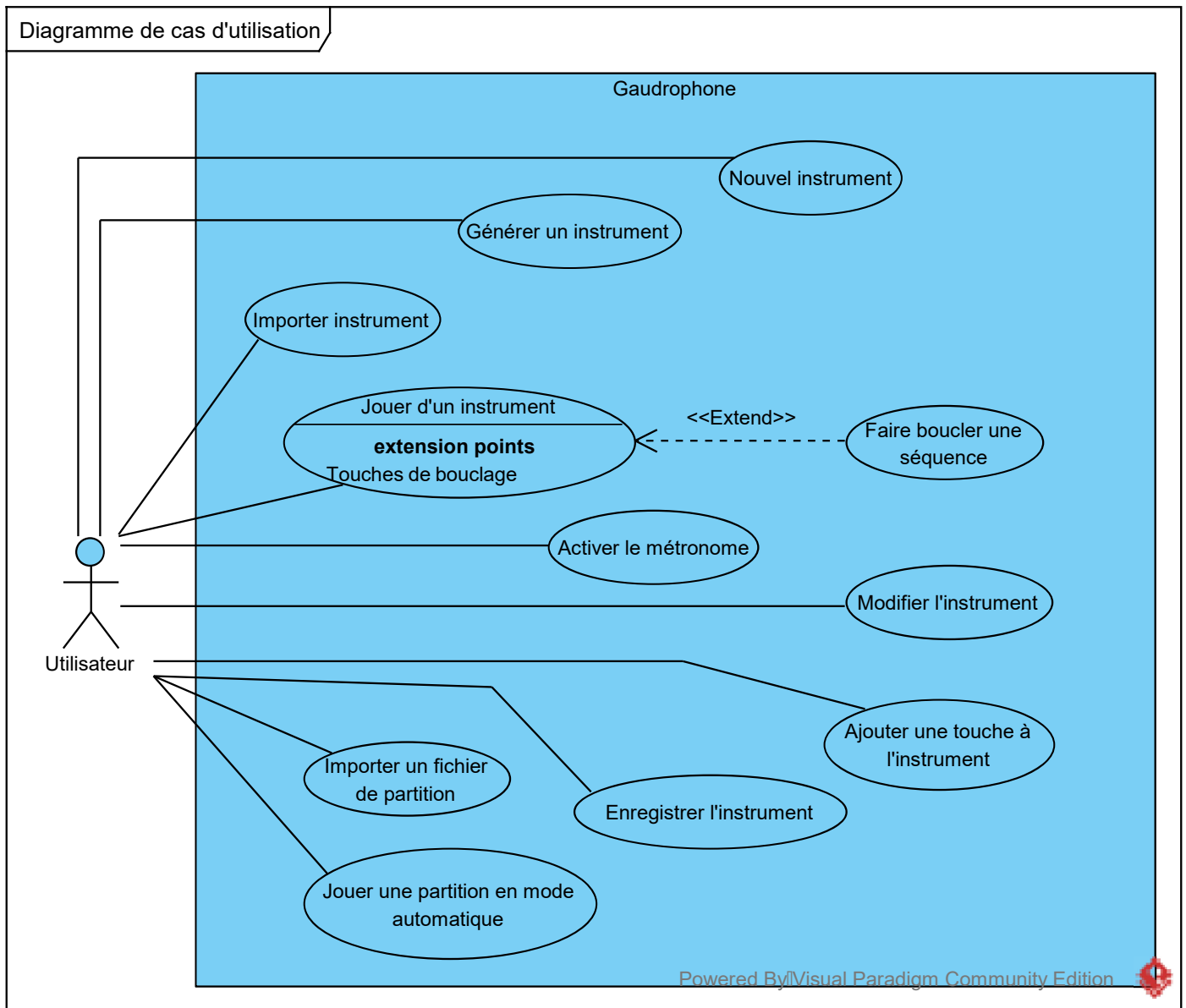
6.1.5 Autres exigences et contraintes

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.

6.2 Modèle du domaine



6.3 Modèle des cas d'utilisation



6.4 Glossaire

Mot	Description
Octave	Une octave est l'intervalle séparant deux sons dont la fréquence fondamentale du plus aigu est le double de celle du plus grave [Wikipedia, 2016a].
Partition	Une partition de musique est un document (en papier, en parchemin, ou en format électronique) qui porte la transcription d'une œuvre musicale [Wikipedia, 2016b].
Touche	Bouton produisant un son lorsqu'il est appuyé.
Fréquence	C'est le nombre d'oscillation d'une onde par seconde, elle s'exprime en hertz. La fréquence d'un son définit sa hauteur et sous-entend une notion de périodicité. Est qualifié de périodique un mouvement ou une oscillation qui se reproduit identiquement à lui-même un certain nombre de fois par seconde [des Gens de Lettres, 2016].
Persistance	Fait de persister, de durer [Larousse, 2016a]
Tonalité	Ensemble des relations entre les degrés hiérarchisés d'une échelle de sons ou d'une gamme, par rapport à la tonique [Larousse, 2016b].
Timbre	Le timbre désigne la différence en son entre deux notes de même fréquence. e.g. deux instruments font la même note avec des timbres différents.

6.5 Note Midi

MIDI Note Numbers for Different Octaves

Octave	Note Numbers											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
0	0	1	2	3	4	5	6	7	8	9	10	11
1	12	13	14	15	16	17	18	19	20	21	22	23
2	24	25	26	27	28	29	30	31	32	33	34	35
3	36	37	38	39	40	41	42	43	44	45	46	47
4	48	49	50	51	52	53	54	55	56	57	58	59
5	60	61	62	63	64	65	66	67	68	69	70	71
6	72	73	74	75	76	77	78	79	80	81	82	83
7	84	85	86	87	88	89	90	91	92	93	94	95
8	96	97	98	99	100	101	102	103	104	105	106	107
9	108	109	110	111	112	113	114	115	116	117	118	119
10	120	121	122	123	124	125	126	127				

7 Bibliographie

Société des Gens de Lettres. musicmot. In *Fréquence*, avril 2016.

Larousse. Larousse. In *Persistence*, avril 2016a.

Larousse. Larousse. In *Tonalité*, avril 2016b.

Contributeurs Wikipedia. Wikipedia, l'encyclopédie libre. In *Octave (musique)*, avril 2016a.

Contributeurs Wikipedia. Wikipedia, l'encyclopédie libre. In *Partition (musique)*, avril 2016b.