

Génie logiciel orienté objet

GLO-2004
Session d'automne 2017

Livrable 4

Destinataire

Jonathan Gaudreault

Présenté par l'équipe Marley

Composé de

Frédéric Pineault

frederick.pineault.1@ulaval.ca

Jean-Philippe Bélanger

jean-philippe.belanger.12@ulaval.ca

Luca Blanchout

Luca.blanchout.1@ulaval.ca

Simon Beaudoin

simon.beaudoin.7@ulaval.ca

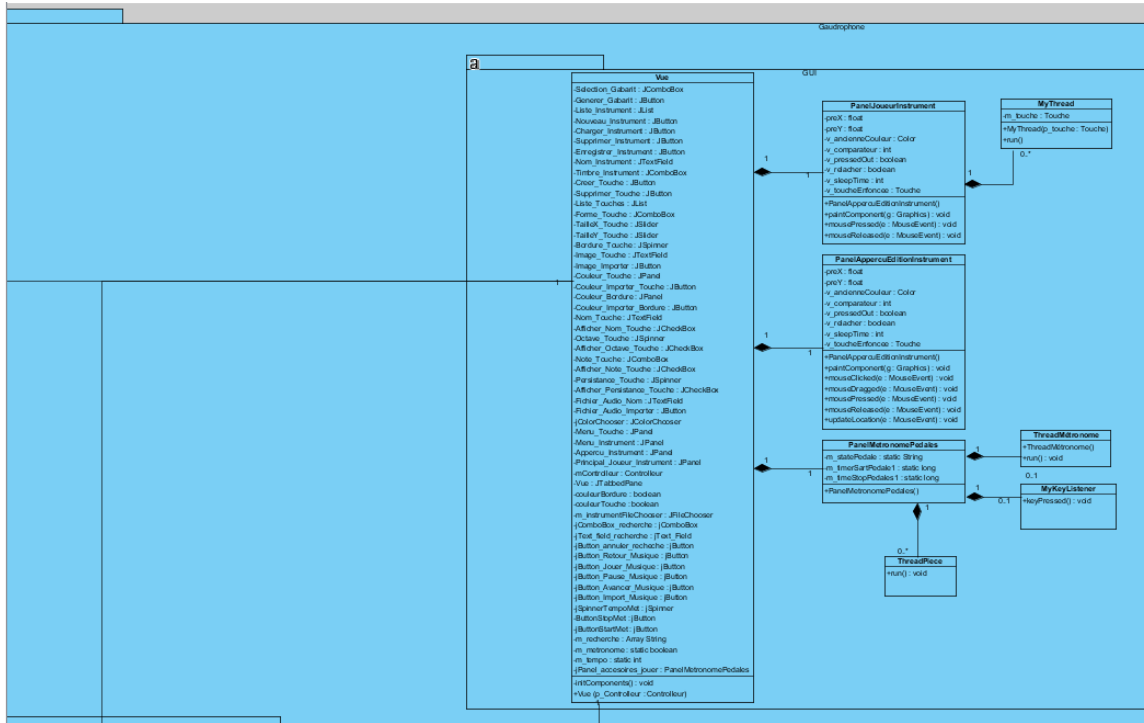
Énoncé de vision

L'équipe Marley est fière de présenter son tout dernier projet : Le Gaudrophone. Il s'agit d'une application développée pour la firme Gaudreault-Lanois ayant pour but de permettre la création d'instrument de musique et de rendre cette création accessible à un très grand public principalement par sa simplicité d'utilisation. Tout utilisateur sera en mesure de créer son propre instrument personnalisé et de le maîtriser à la perfection. L'utilisateur pourra donc créer des touches et les designer pour qu'elles répondent à ces besoins. L'ensemble de ces touches composeront donc l'instrument.

En plus de permettre la création de différents instruments, il est aussi possible de se pratiquer avec ceux-ci. L'utilisateur peut en premier lieu se pratiquer par lui-même en jouant des pièces qu'il connaît. Une autre façon de se pratiquer est avec la lecture de pièce musicale. Cette option permet de suivre une pièce musicale où l'utilisateur doit appuyer sur les touches au moment où elles sont illuminées pour jouer le son de la note et donc de jouer une mélodie. Le futur musicien peut donc apprendre les pièces avec une grande facilité et ainsi devenir encore plus performant avec son instrument. Un métronome sera toujours disponible pour les utilisateurs afin de se pratiquer à conserver le rythme musical. Le bouclage en direct de séquences de notes permet aux musiciens expérimentés d'explorer leur côté créatif et ainsi de les aider à produire leur propre composition. Il est également possible de jouer une pièce avec le son et de simplement écouter la mélodie pour relaxer.

Le Gaudrophone est un projet innovateur dans le domaine musical puisque celui-ci offre un accès immédiat à plusieurs instruments de musique. Il permet aussi à n'importe qui de devenir un excellent musicien grâce à ses options de pratique élaborées.

Diagramme de classe de conception



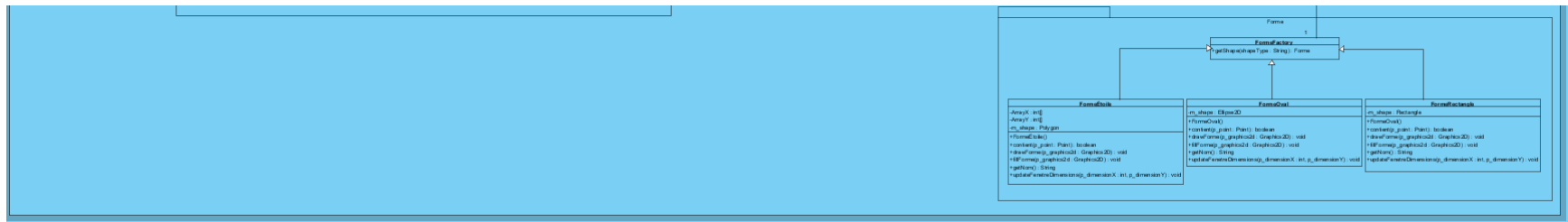
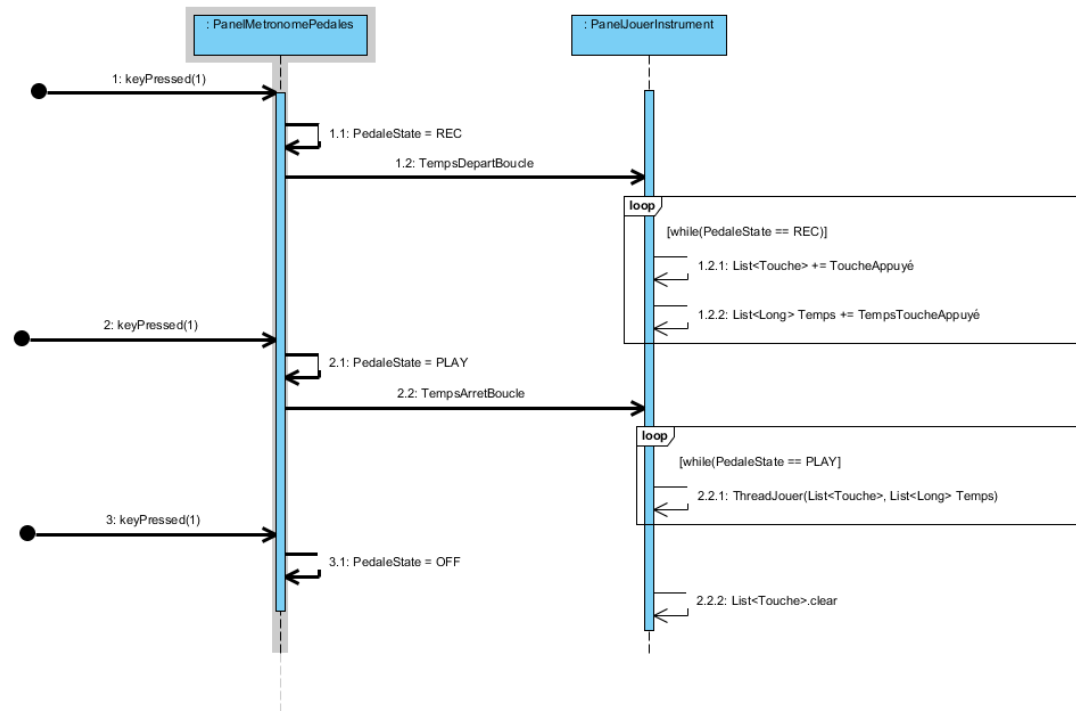


Diagramme de séquence live-looping



Texte explicatif live-looping

Le fonctionnement du live looping est le même pour toutes les pédales. Lorsque l'on enfonce la touche associée à la pédale (1 à 9 sur le clavier), l'état de la pédale change. Il y a 3 états possible : OFF, REC et PLAY. Les pédales sont initialisés à l'état OFF. Lorsque l'on change d'état pour REC, on note le temps de départ. Ensuite, tant et aussi longtemps que l'on reste dans cet état, les touches utilisées sont placées dans un ArrayList de touche. On place aussi le temps qu'elles sont appuyées dans un ArrayList de Long. La touche aura le même index que son temps dans les ArrayLists. Lorsque l'on change d'état pour PLAY. On note le temps de fin. On démarre un thread qui s'occupe de jouer les notes au bon temps dans une boucle en fonction du temps de début et du temps de fin. La boucle se répète jusqu'à changement d'état. Lorsque l'on change d'état pour OFF. La boucle arrête de jouer et les ArrayList sont effacés.

Conclusion

Notre application répond donc aux exigences du client. Certains points forts ressortent de notre application. En premier lieu, sa simplicité permet à n'importe qui de prendre en main le logiciel et de l'utiliser rapidement. Notre application offre beaucoup d'options comme le fait de nommer une touche par le nom qu'on désire. Les changements appliqués aux touches sont également directement appliqués sans besoin de passer par l'intermédiaire d'un bouton du type « appliquer ». Il est également possible d'avoir plusieurs instruments dans notre application et de changer d'instruments sans avoir à en charger un nouveau à chaque fois. Le changement d'instruments se fait donc beaucoup plus rapidement.

Certains points pourraient cependant être améliorés pour rendre notre application plus performante et plus « *user friendly* ». Par exemple, il serait possible de mieux présenter le live-looping en démontrant dans quel état nous sommes rendus. Présentement, l'utilisateur doit savoir par lui-même dans quel état il est rendu. Il serait également possible d'améliorer les bordures pour les rendre indépendantes puisque dans notre application, la largeur des bordures sont dépendantes l'une de l'autre.

Contribution

Luca Blanchout : Barre de recherche, création de fonctions de soutiens pour les autres membres de l'équipe, rapport.

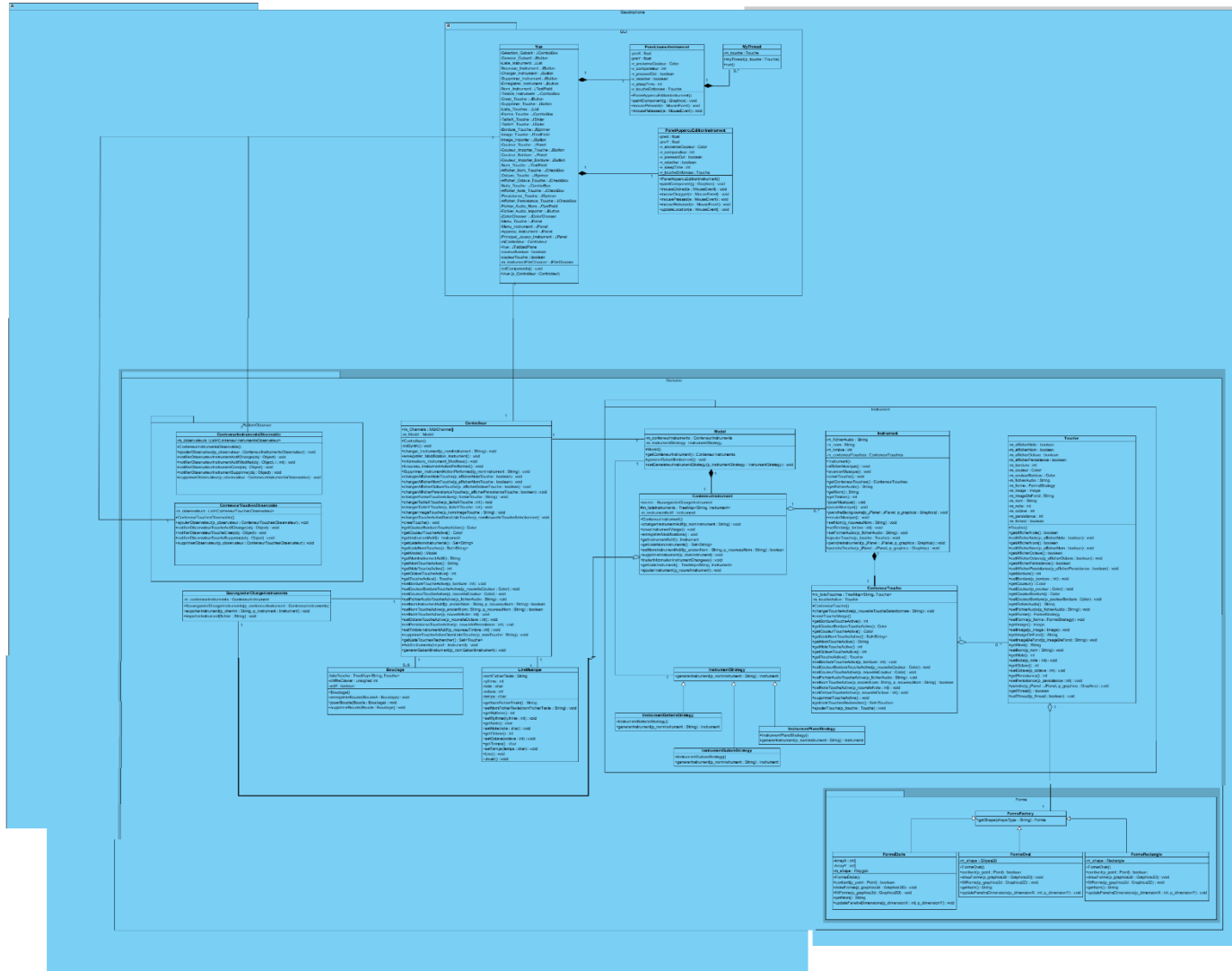
Jean-Philippe Bélanger : Diagramme de conception, travail sur la lecture de fichiers .txt, création de fonctions de soutiens pour les autres membres de l'équipe

Frédéric Pineault : Live-looping, diagrammes de séquence pour le live-looping, texte explicatif du live-looping.

Simon Beaudoin : Travail sur la lecture des fichiers txt, création du module pour jouer les pièces musicales avec ou sans son.

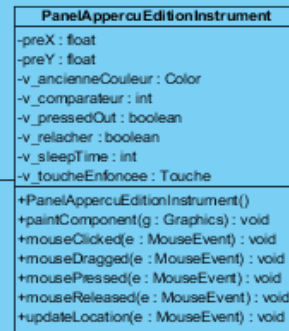
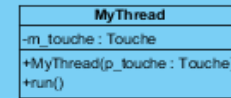
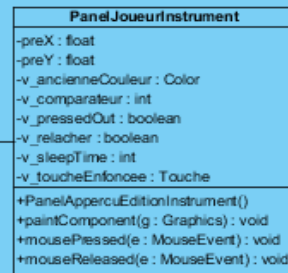
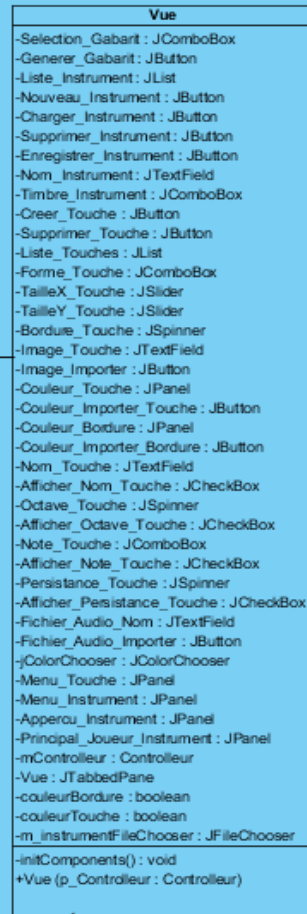
Livable 3

Diagramme de classe de conception



a

GUI



1

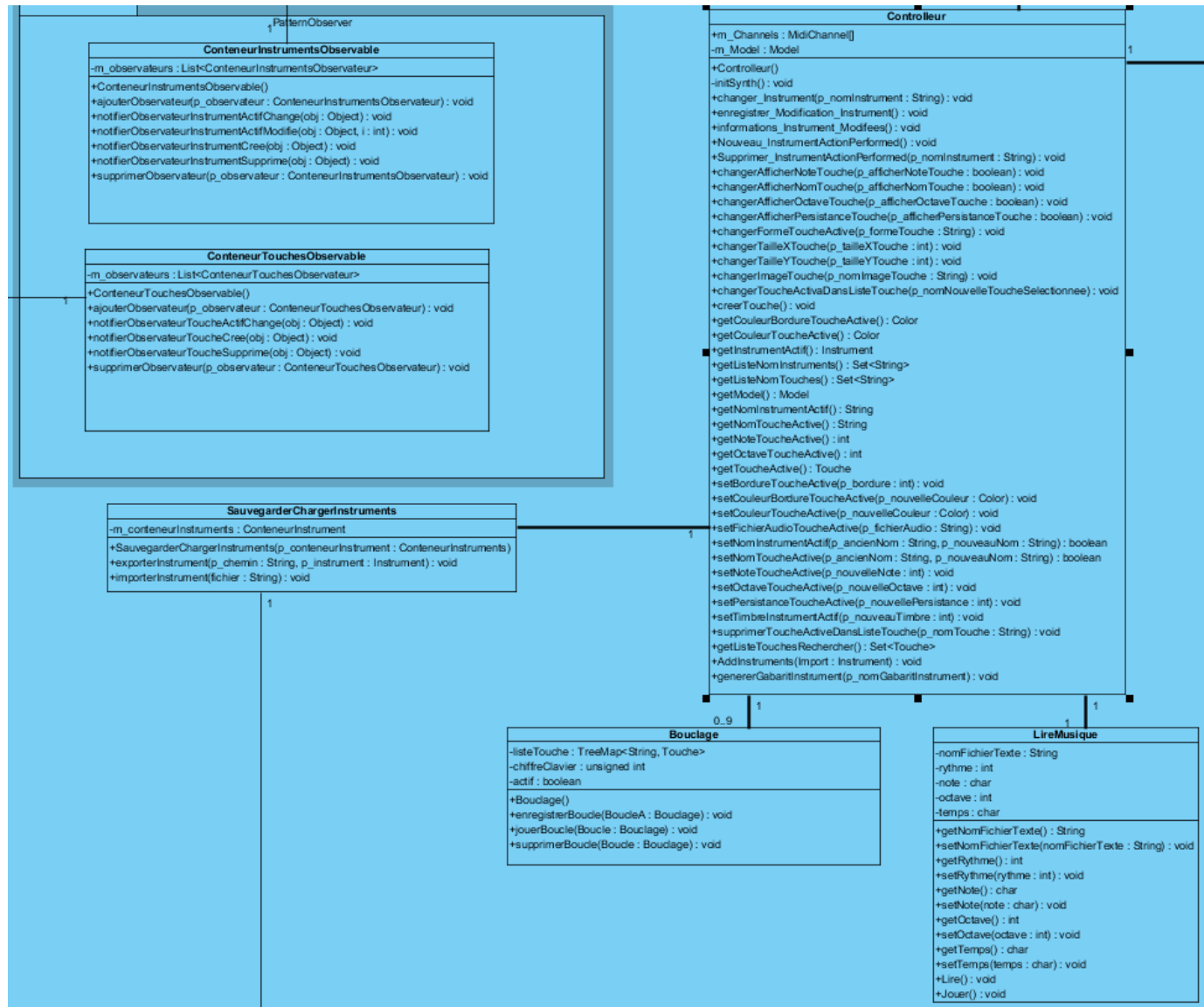
1

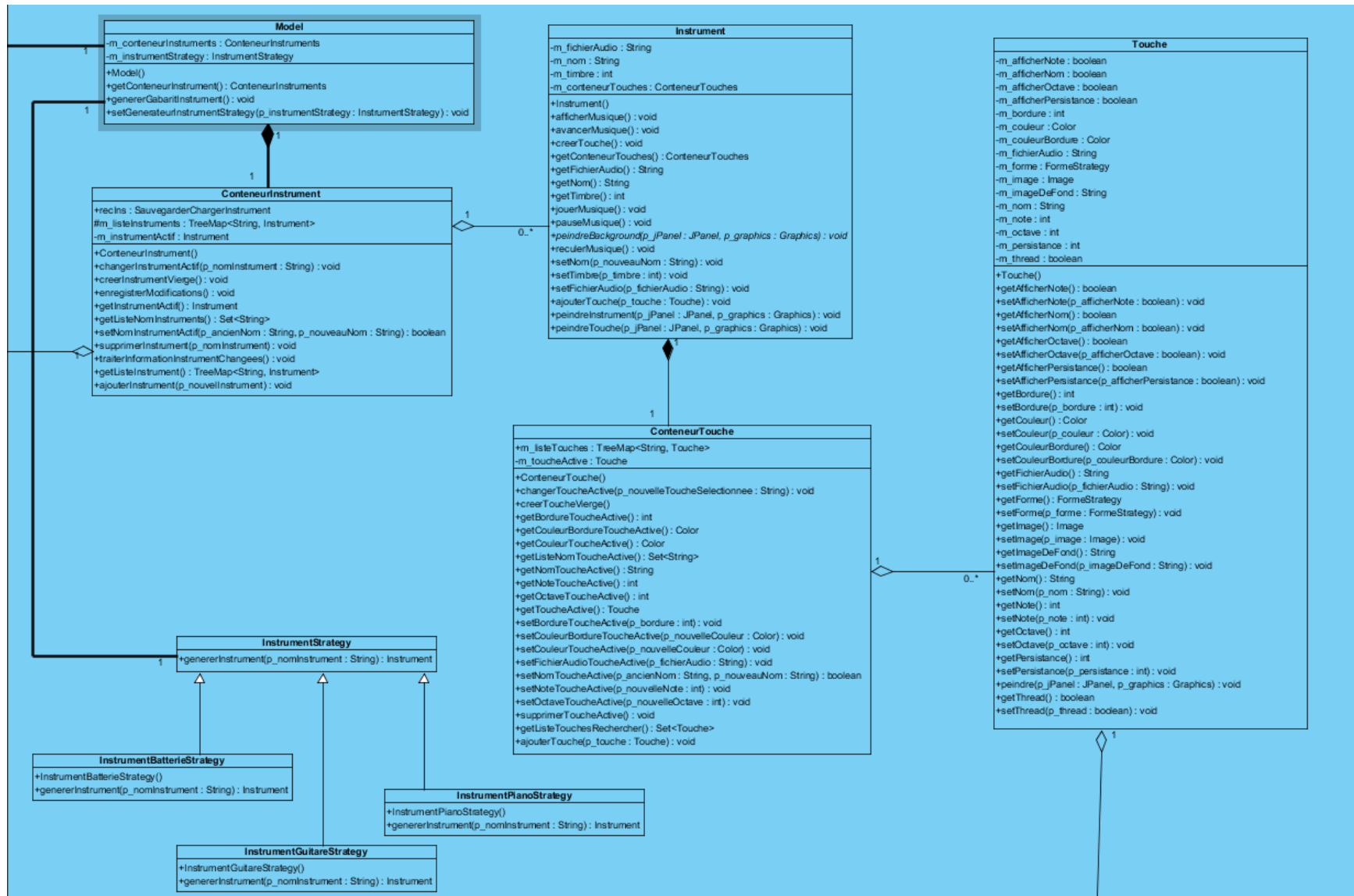
1

0..*

1

1





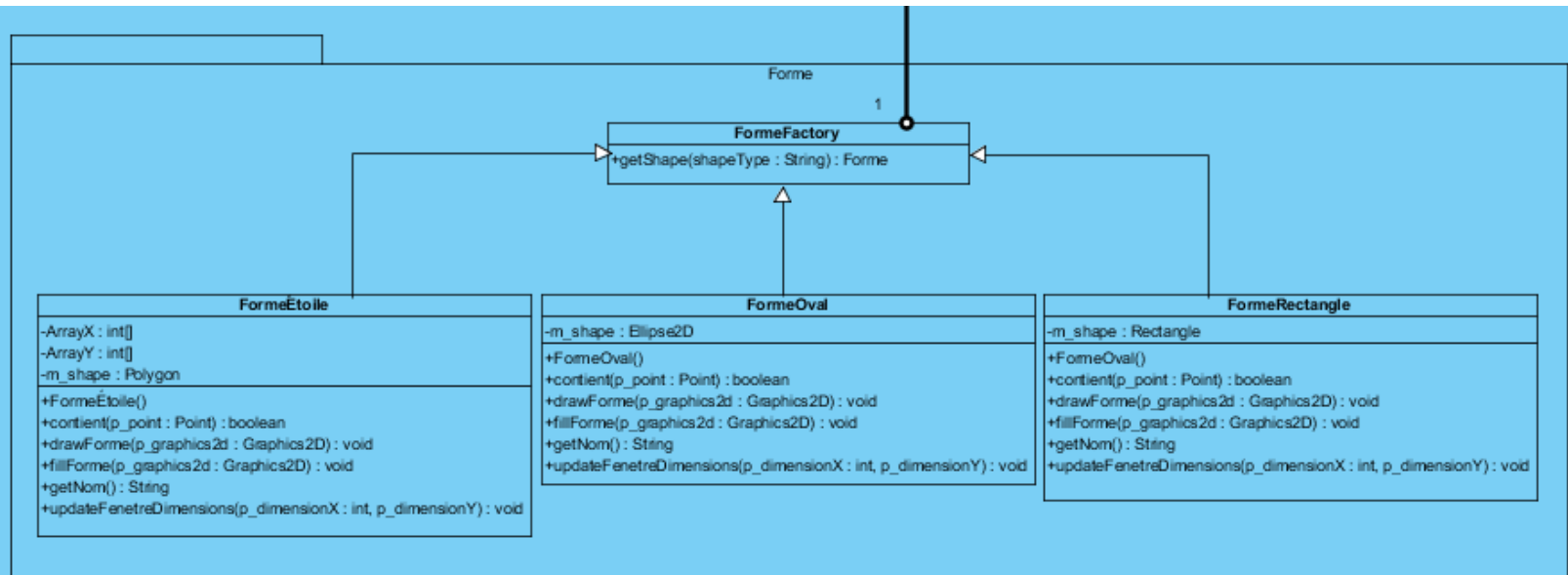
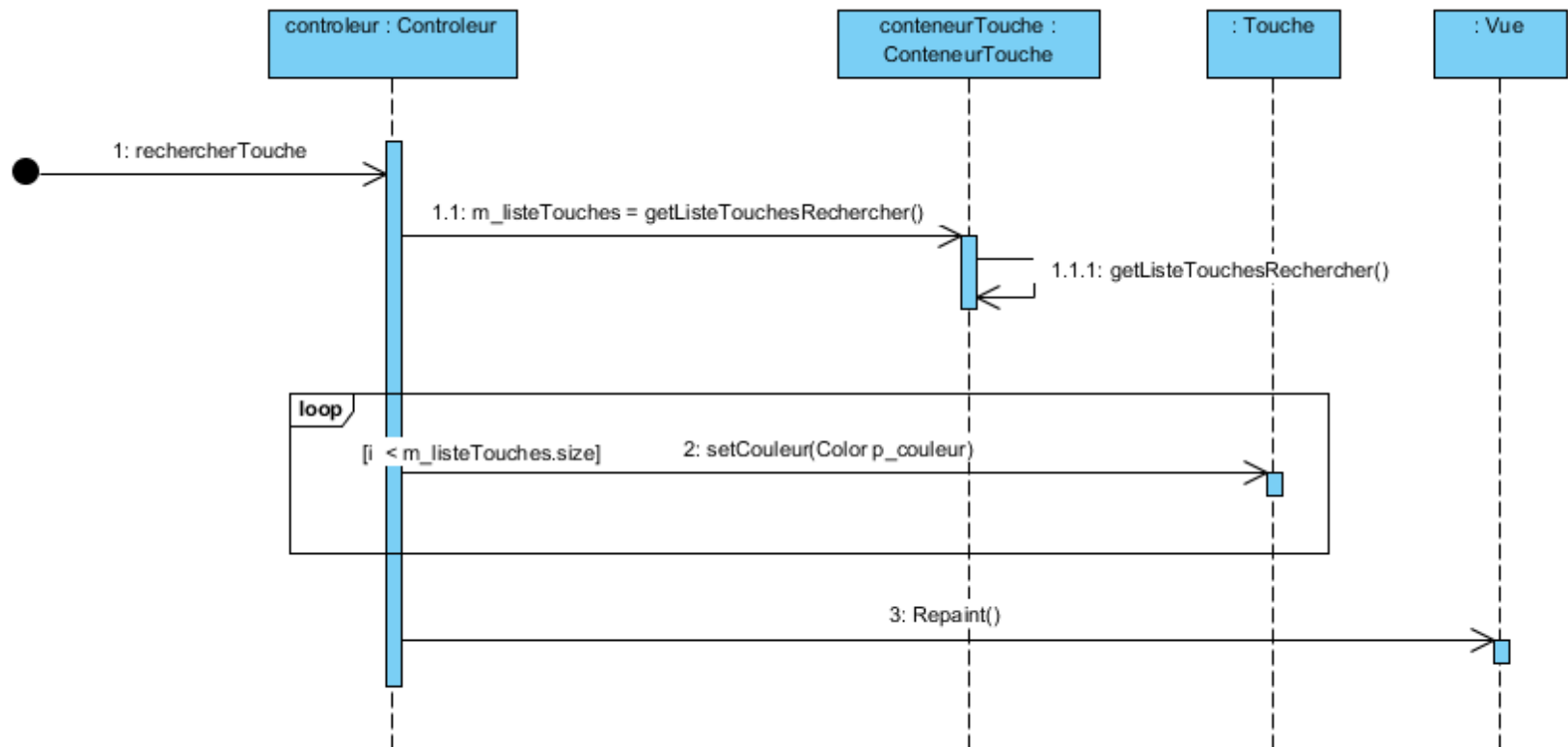


Diagramme de séquence de la barre de recherche

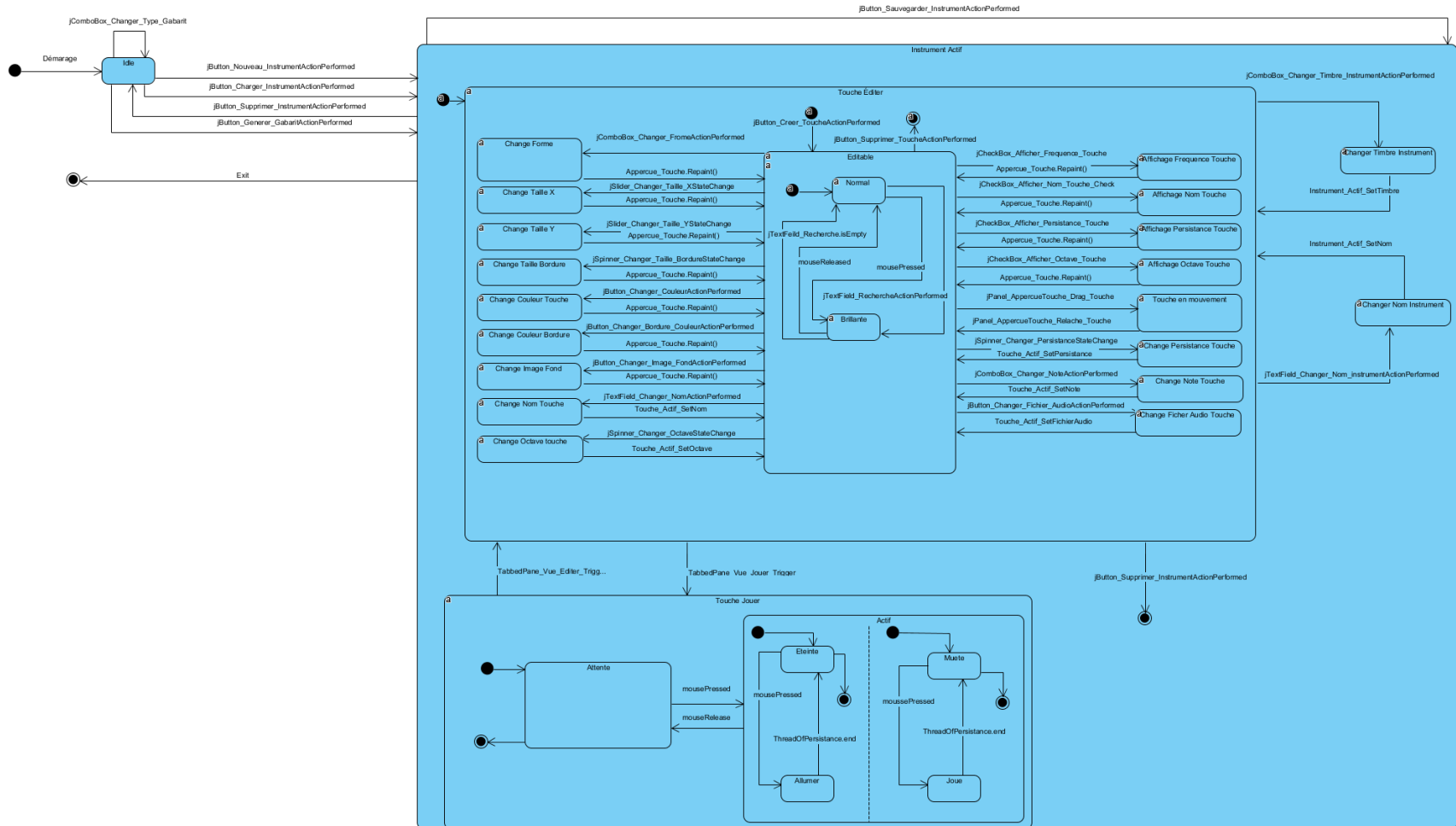


Lorsque l'utilisateur écrit dans la boîte de recherche, le texte qui y est écrit est envoyé au contrôleur. Ce string est par la suite redirigé vers le conteneur de touches.

Le conteneur de touches cherche les touches correspondantes grâce au TreeMap implémenté qui contient toutes les touches créées. Leurs informations seront analysées grâce à leur *getter* et les touches correspondantes à la recherche seront retournés dans un conteneur. Le catalogue renvoie au contrôleur le conteneur. Un itérateur s'occupera d'itérer sur ce conteneur pour changer la couleur de chaque touche présente. Cela permettra donc de fournir une référence visuelle à l'utilisateur pour identifier les touches correspondantes à la recherche.

Une fois que l'itérateur a fini de changer les couleurs des touches retournées, le JPanel est repeint pour afficher les changements.

Diagramme d'états



Contribution

Luca Blanchout : Créer éditer un instrument, jouer l'instrument, importer un fichier audio, s'assurer qu'il n'y ait pas de boîte flottante, gabarit.

Simon Beaudoin : Base du code ainsi que le look de l'application, sérialisation, créer et éditer un instrument, gestion de la redimensionnable, éléments sur les touches.

Frédéric Pineault : Gestion du son agréable, jouer l'instrument, gabarit, diagramme de classe de conception et diagramme de séquence

Jean-Philippe Bélanger : FormeFactory, sérialisation, diagramme d'état et de séquence.

Rapport Livrable 2

Diagramme de classe de conception

Texte explicatif pour les classes

FenetrePrincipale : Cette classe est la fenêtre principale de l'application. Elle contient 3 onglets différents qui permettent de changer parmi les différentes fonctionnalités de notre application.

PanneauEditeurTouche : Cette classe fait partie du GUI et permet d'afficher toutes les fonctionnalités disponibles pour créer et éditer les touches. Chaque événement est ensuite redirigé vers le contrôleur.

PanneauEditeurInstrument : Cette classe fait partie du GUI et permet d'afficher toutes les fonctionnalités disponibles pour créer et éditer les instruments. Chaque événement est ensuite redirigé vers le contrôleur.

PanneauJouerInstrument : Cette classe fait partie du GUI et permet d'afficher toutes les fonctionnalités disponibles pour jouer d'un instrument. Chaque événement est ensuite redirigé vers le contrôleur.

PanneauDessin : Cette classe fait partie du GUI et hérite de la classe JPanel de Java Swing. Certaines méthodes seront modifiées à l'aide du polymorphisme pour être adaptées à nos besoins. Cette classe est utilisée dans les classes PanneauEditeurTouche et PanneauEditeurInstrument.

Contrôleur : Cette classe représente le contrôleur de Larman. Elle permet de rediriger toutes les fonctions de notre programme. Les événements générés par les classes dans le package UI passent par cette classe et sont redirigés vers les classes respectives dans le package du domaine.

CatalogueTouche : Cette classe représente tous les objets Touche créés. Elle permet par la suite d'être utilisée lors de la création d'instruments pour obtenir tous les objets Touche créés et déterminer lesquels vont être ajoutés à l'instrument. C'est grâce à cette classe qu'il sera possible de chercher les différentes touches à l'aide de la barre de recherche.

Touche : Cette classe représente les touches possibles d'instancier pour placer sur un instrument par la suite. Les objets Touche font partie intégrante des objets Instrument instanciés.

CatalogueInstrument : Cette classe représente tous les objets Instrument créés. Elle permet par la suite d'être utilisée lorsque l'on veut charger ou enregistrer un instrument créé.

Instrument : Cette classe représente les instruments possibles d'instancier. Ces objets sont constitués de touches et de caractéristiques propres à eux-mêmes. Ces objets sont ensuite placés dans un objet CatalogueInstrument.

Forme : Cette classe permet d'instancier des objets Forme composés de différentes coordonnées reliées entre elles. Ces objets sont ensuite placés dans un objet ConteneurForme.

ConteneurForme : Cette classe permet d'emmagasinier les différents objets Forme instanciés. C'est une liste des objets Forme disponibles pour affecter à un objet Touche.

Coordonnees : Cet objet représente un couple x et y représentant une coordonnée cartésienne qui est utilisée par les objets Forme et Instrument.

AfficheurInstrument : Cette classe permet d'afficher un instrument dans les panneaux. Une fois un instrument sélectionné, elle dessine l'instrument et le rend actif.

Bouclage : Cette classe s'occupe d'enregistrer les différents bouclages possibles pour les différentes touches. Les boucles sont soit enregistrées, jouées ou supprimées.

Métronome : Cette classe permet d'activer ou non un métronome. Le métronome génère un son à une fréquence donnée. Cet objet est actif ou non.

IGenerateurInstrumentStrategie : Cette interface permet de donner un corps de base aux classes GenerateurGuitareStrategie, GenerateurPianoStrategie et GenerateurBatterieStrategie. C'est par cette interface que le contrôleur passera pour générer les différents gabarits disponibles. Comme c'est une interface, il est impossible d'instancier un objet de cette classe.

GenerateurGuitareStrategie : Cette classe permet de générer dynamiquement une guitare pour que l'utilisateur puisse en jouer.

GenerateurPianoStrategie : Cette classe permet de générer dynamiquement un piano pour que l'utilisateur puisse en jouer.

GenerateurBatterieStrategie : Cette classe permet de générer dynamiquement une batterie pour que l'utilisateur puisse en jouer.

Diagramme de package

Texte explicatif pour les packages

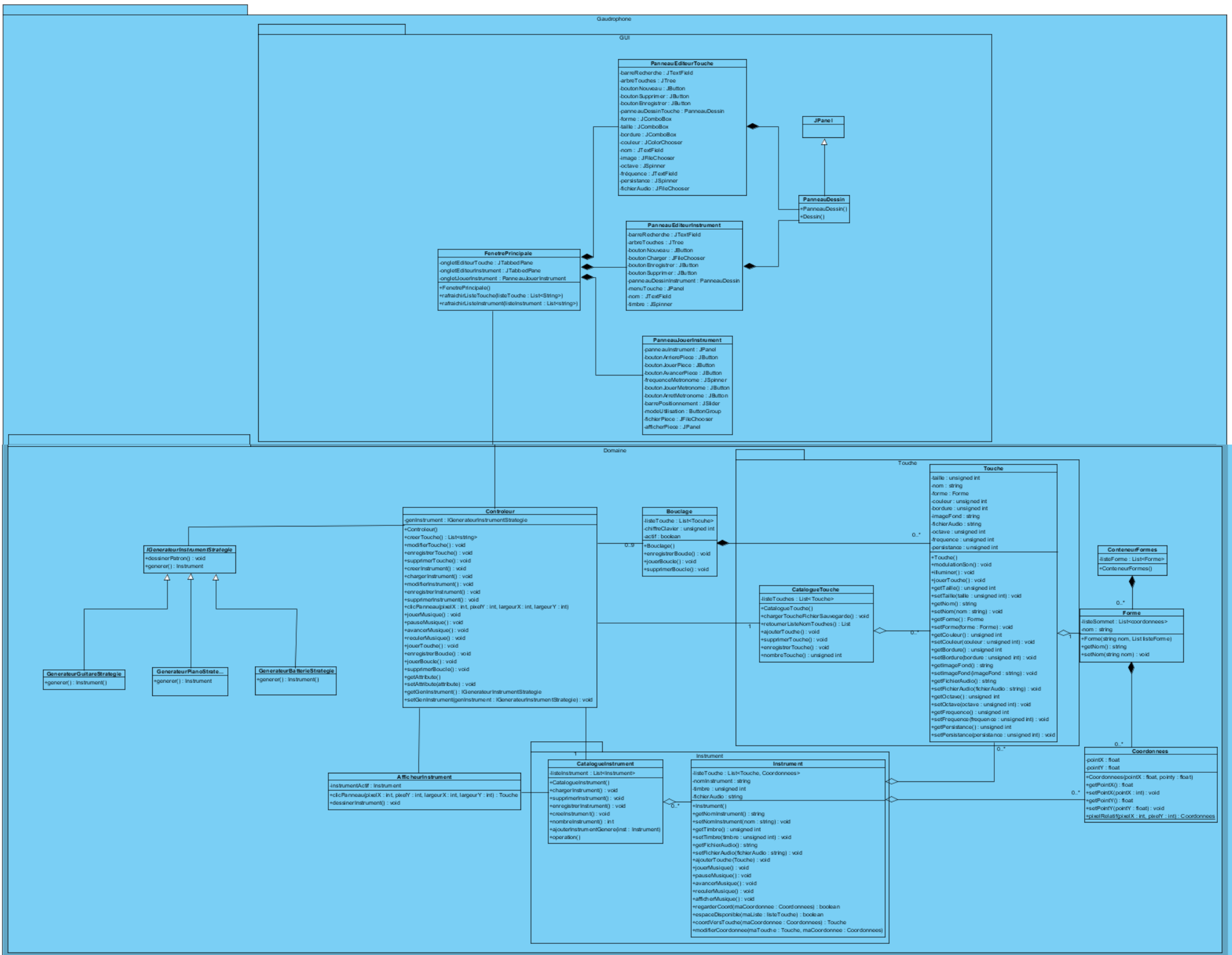
Les packages ont été séparés selon le principe de la séparation modèle-vue de l'architecture logique. Comme cela, on obtient deux ensembles qui peuvent être séparés et tester individuellement.

Package GUI : Ce package représente l'interface graphique du programme. Les différents éléments créent des événements lorsqu'ils sont utilisés et ces événements sont redirigés vers les contrôleurs, qui est lui-même dans le package du domaine. Les classes FenetrePrincipale, PanneauEditeurTouche, PanneauEditeurInstrument, PanneauJouerInstrument et PanneauDessin font partie de ce package.

Package Domaine : Ce package représente toute la logique présente derrière le GUI. Le contrôleur reçoit tous les événements provenant de l'interface utilisateur et les redirige vers les classes respectives. Les classes Controleur, CatalogueTouche, Touche, CatalogueInstrument, Instrument, Forme, ConteneurForme, Coordonnees, AfficheurInstrument, Bouclage, Métronome, IGenerateurInstrumentStrategie, GenerateurGuitareStrategie, GenerateurPianoStrategie et GenerateurBatterieStrategie font partie de ce package.

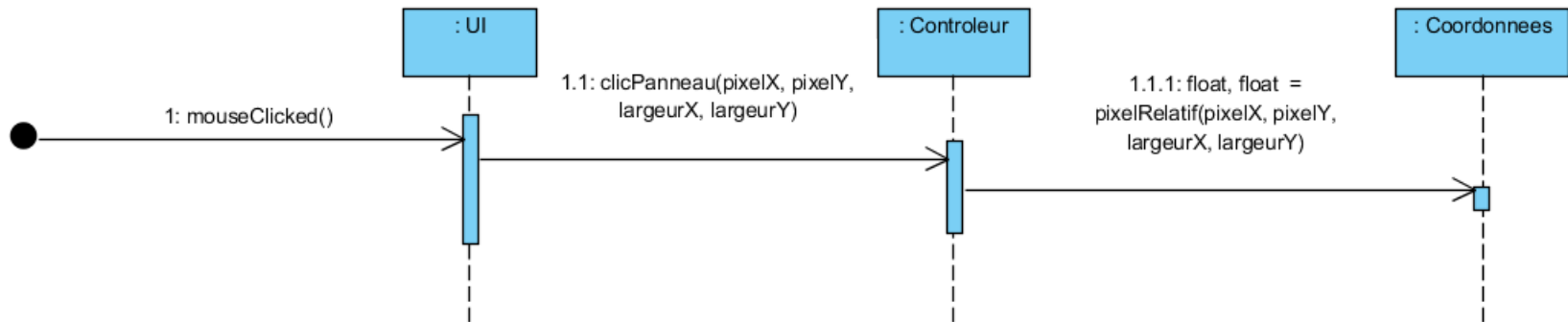
Package Instrument : Ce package représente les classes associées à l'instrument. Les classes présentes dans ce package permettent de créer un instrument et de l'enregistrer. Les classes sont Instrument et CatalogueInstrument. Ce package fait partie du package Domaine.

Package Touche : Ce package représente les classes associées aux touches. Les classes présentes dans ce package permettent de créer une touche et de l'enregistrer. Les classes sont Touche et CatalogueTouche. Ce package fait partie du package Domaine.



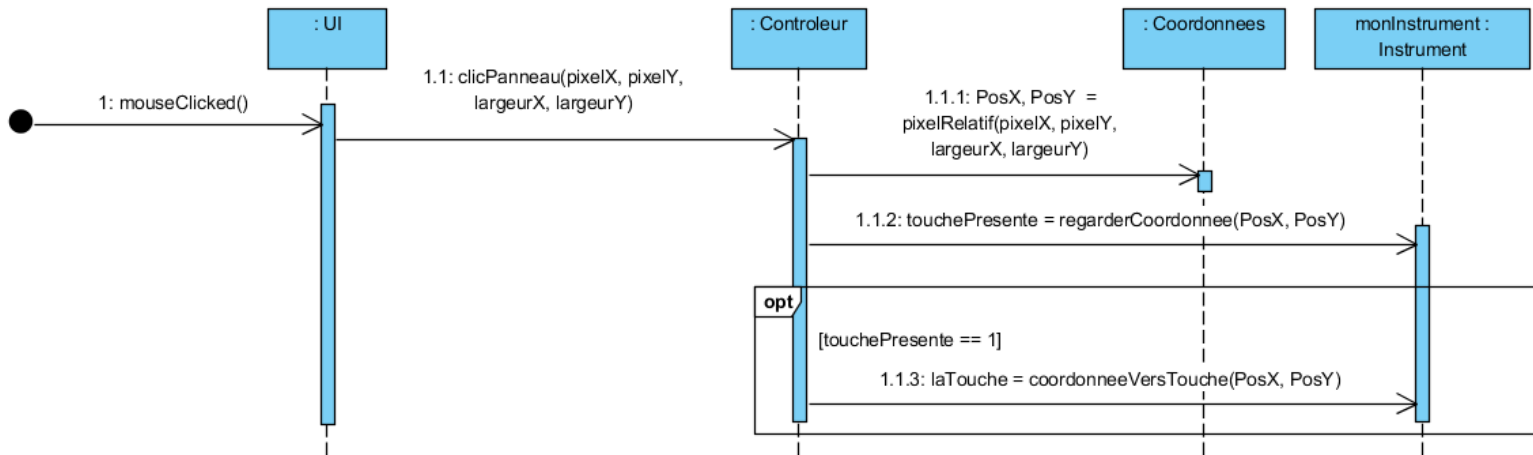
Diagrammes de séquence de conception

3.1 - Conversion entre les pixels et les valeurs relatives



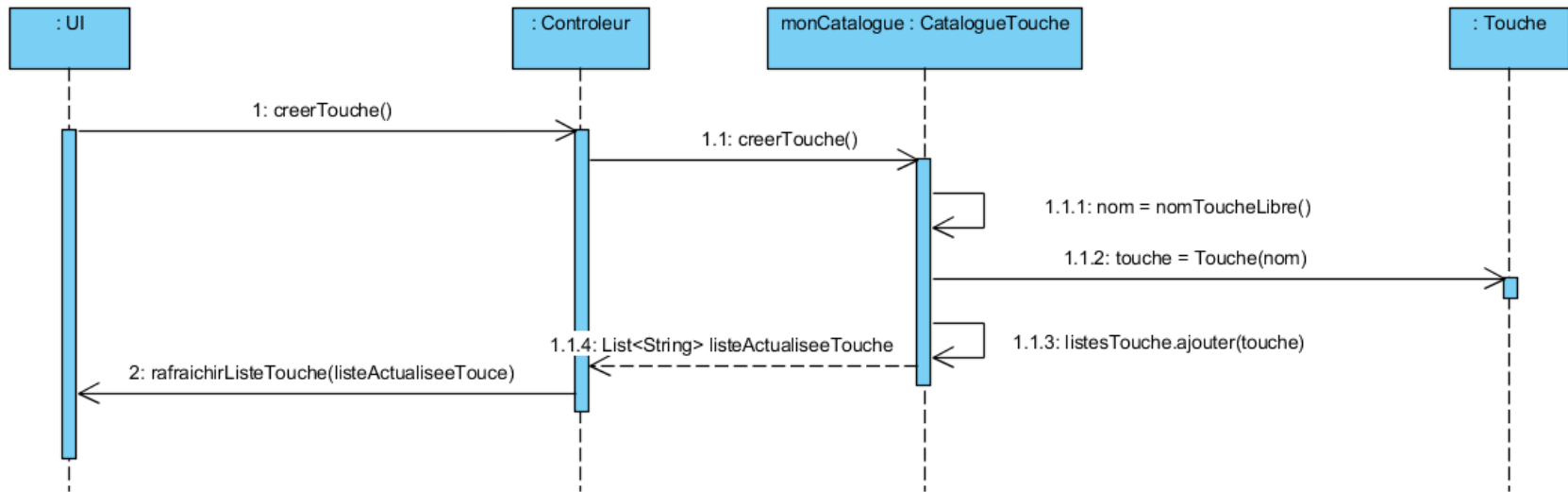
Le concept des pixels pour notre application s'applique seulement dans la couche utilisateur, par exemple, lorsqu'on va chercher la position de la souris ou pour définir la largeur de notre application sur l'écran de l'utilisateur. La couche du domaine quand à elle peut recevoir des pixels et elle va directement les transformés en coordonnée. Dans notre schéma, le contrôleur reçoit 4 données en pixel. Dans notre exemple, c'est la position de la souris lorsqu'il a un clic ainsi que la longueur et la largeur de la fenêtre principale. Il appelle une fonction statique pour produire les coordonnées de la pointe de la souris.

3.2 - Déterminer le clic



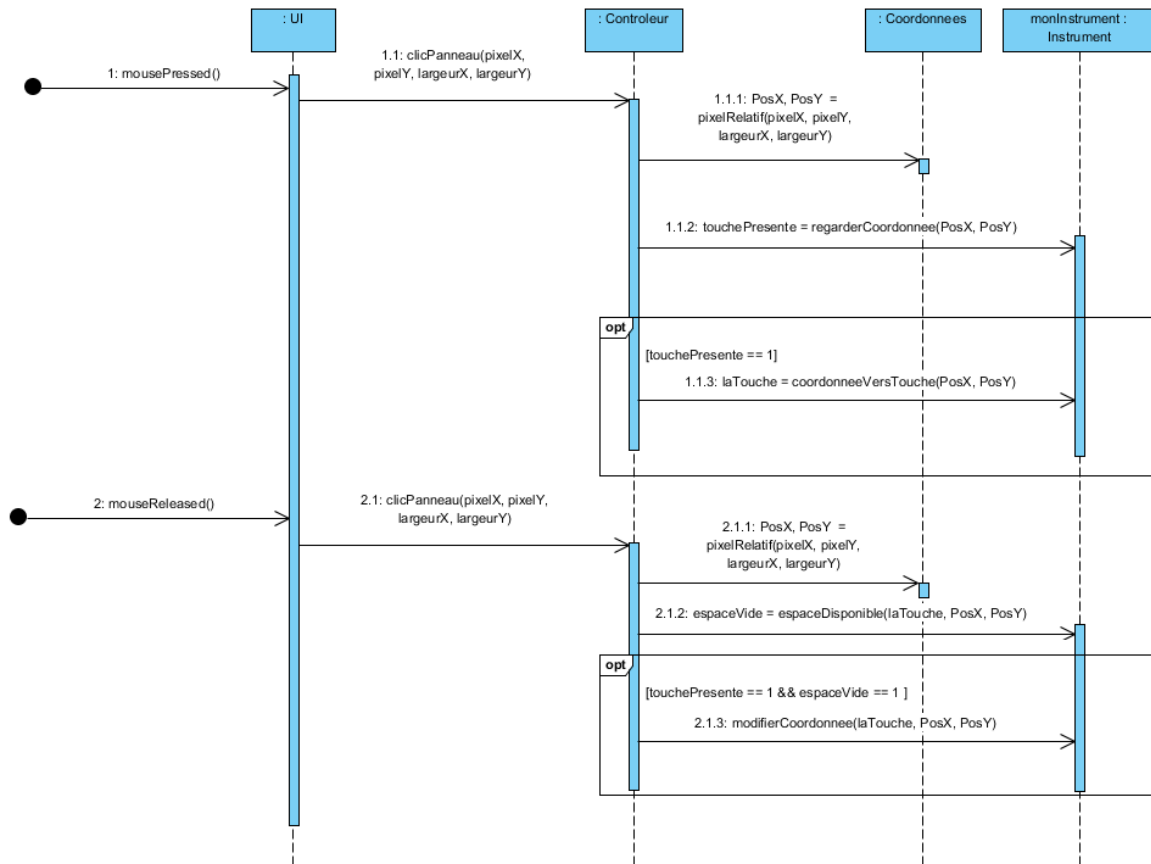
Le principe de départ est similaire à l'exemple donné au diagramme présenté en 3.1, on reçoit le clic et on va chercher les coordonnées de celui-ci. Ensuite, la fonction regarderCoordonnee est appelée. Elle cherche à déterminer si le clic de la souris se retrouve sur une touche. Si c'est le cas, il est donc possible de savoir quelle est la touche en question.

3.3 - Ajout d'une touche



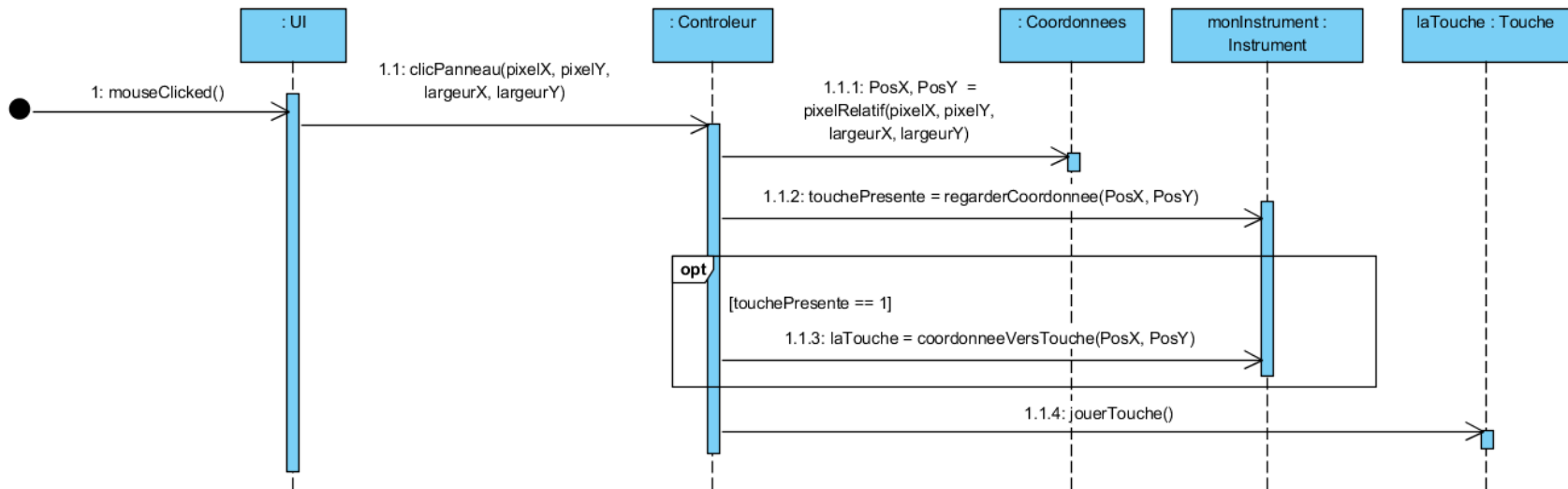
Lorsque l'utilisateur clic sur le bouton pour créer une nouvelle touche, le contrôleur est appelé. Celui-ci appelle la fonction 'creerTouche' du catalogue de touche. Cette dernière fonction se charge de trouver un nom libre du style 'NouvelleTouche1' ou 'NouvelleTouche2', etc. Une fois le nom par défaut de la nouvelle touche trouvé, le constructeur de Touche par défaut est appelé et le nom de la nouvelle touche lui est assigné. Le catalogue de touche va ensuite ajouter cette nouvelle touche à sa liste contenant toutes les touches. Finalement, la fonction 'creerTouche' va retourner une liste de string représentant le nom de toutes les touches qui reflètera l'ajout de la touche. En recevant cette liste de string, le controlleur mettra à jour `la liste de touche au niveau de l'interface utilisateur via la fonction 'rafraichirListeTouche'.

3.4 - Déplacer une touche



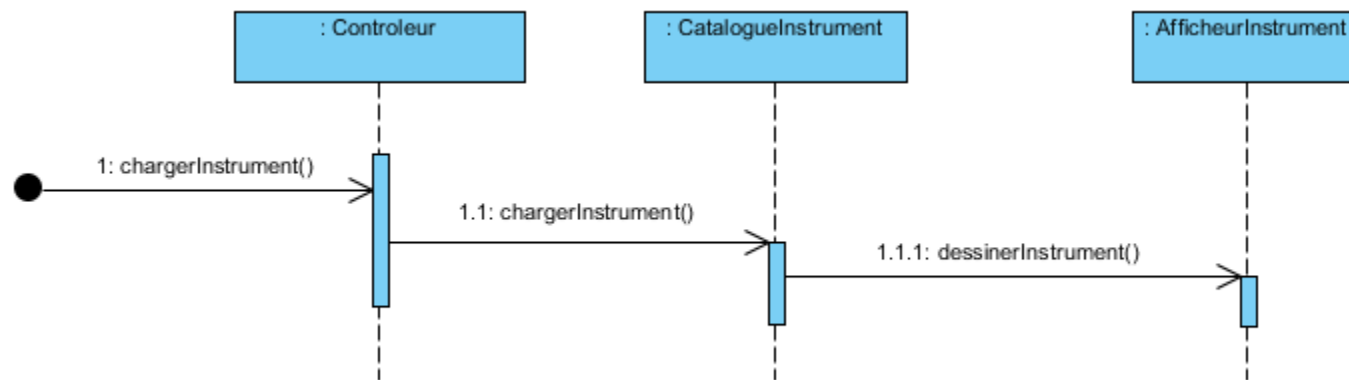
Le principe de départ est similaire au diagramme présenté en 3.2, lorsque la souris est maintenue par l'utilisateur, on va chercher la touche qui est appuyé par la souris, si il y en a une bien entendu. Ensuite, lorsque la souris est relâchée, on regarde où est la souris. On vérifie que la touche rentre bel et bien dans l'espace où l'on veut la déplacer. Si tout est possible, on modifie donc les coordonnées de la touche et ainsi le déplacement à lieu.

3.5 - Produire le son



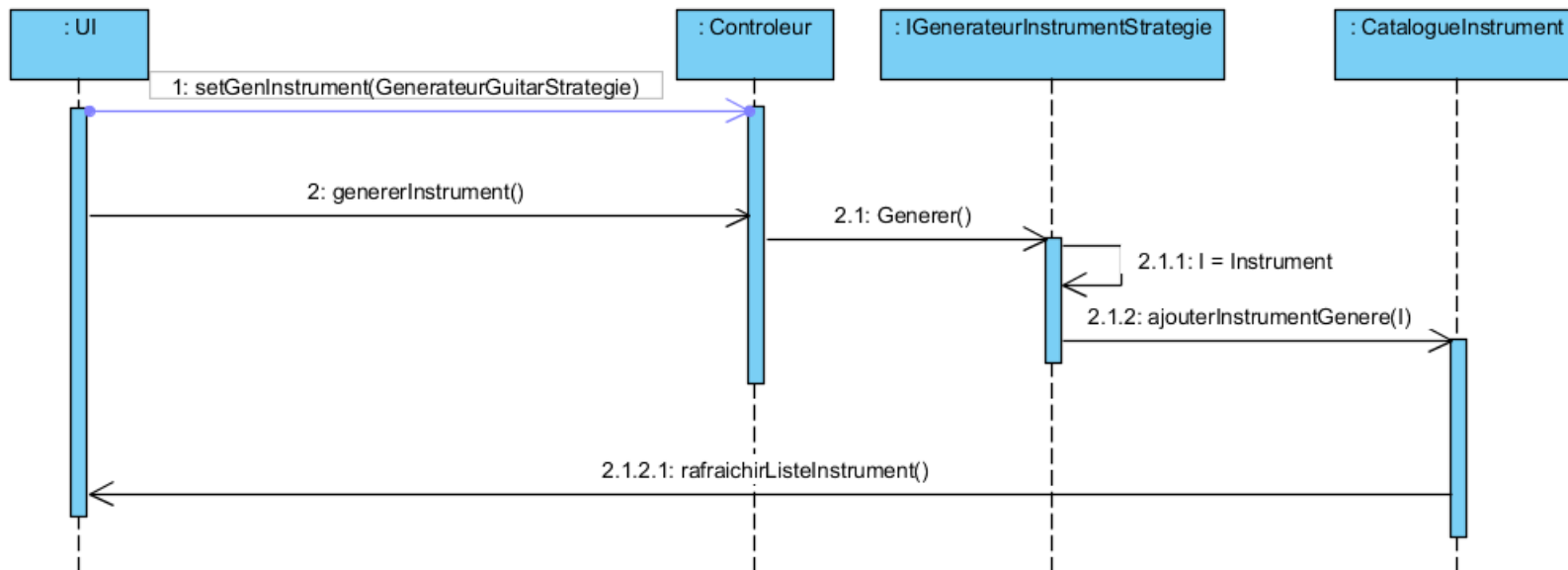
Lorsqu'un clic sur le panneau de l'instrument survient dans l'onglet jouer instrument, l'UI mémorise la position de la souris et il demande au contrôleur d'utiliser les informations pour trouver à quel endroit le clic a eu lieu. Le contrôleur prend les informations et fait le tour de la liste des touches dans CatalogueTouche pour savoir si une touche a été cliqué. S'il trouve une correspondance, il fait jouer le son de la touche.

3.6 - Afficher l'instrument



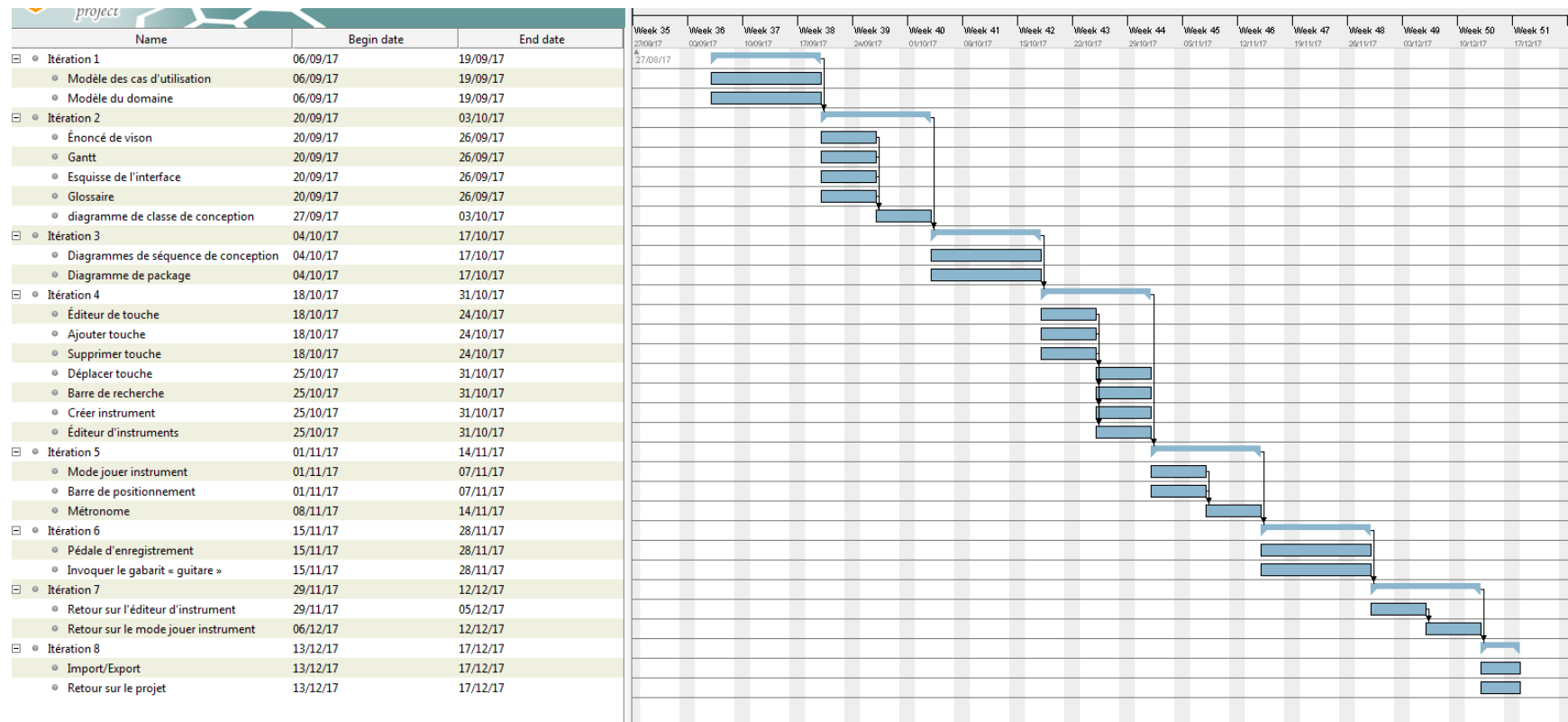
L'utilisateur choisit de charger son instrument, le contrôleur va donc demander au catalogue de charger celui-ci. L'afficheur d'instrument s'occupe de placer les touches au bon endroit.

3.7 - Génération instrument



Dans un premier temps, l'utilisateur sélectionne le gabarit de l'instrument à générer, en accordance avec le pattern Strategy. Dans un second temps, lorsque l'utilisateur décide d'appuyer sur le bouton pour générer l'instrument donc il a préalablement muté le gabarit, le contrôleur appelle la fonction virtuelle 'generer' de l'interface 'GenerateurInstrumentStrategie'. Cette fonction, implémentée dans 'GenerateurGuitarStrategie' se charge de créer un instrument de type guitar de toute pièce. Cet instrument est ensuite ajouté au catalogue d'instrument via la fonction 'ajouterInstrumentGenere'. Une fois l'instrument ajouté au catalogue, ce dernier va ensuite actualiser la liste des instruments au niveau de l'interface utilisateur pour témoigner de la création de l'instrument.

Gantt



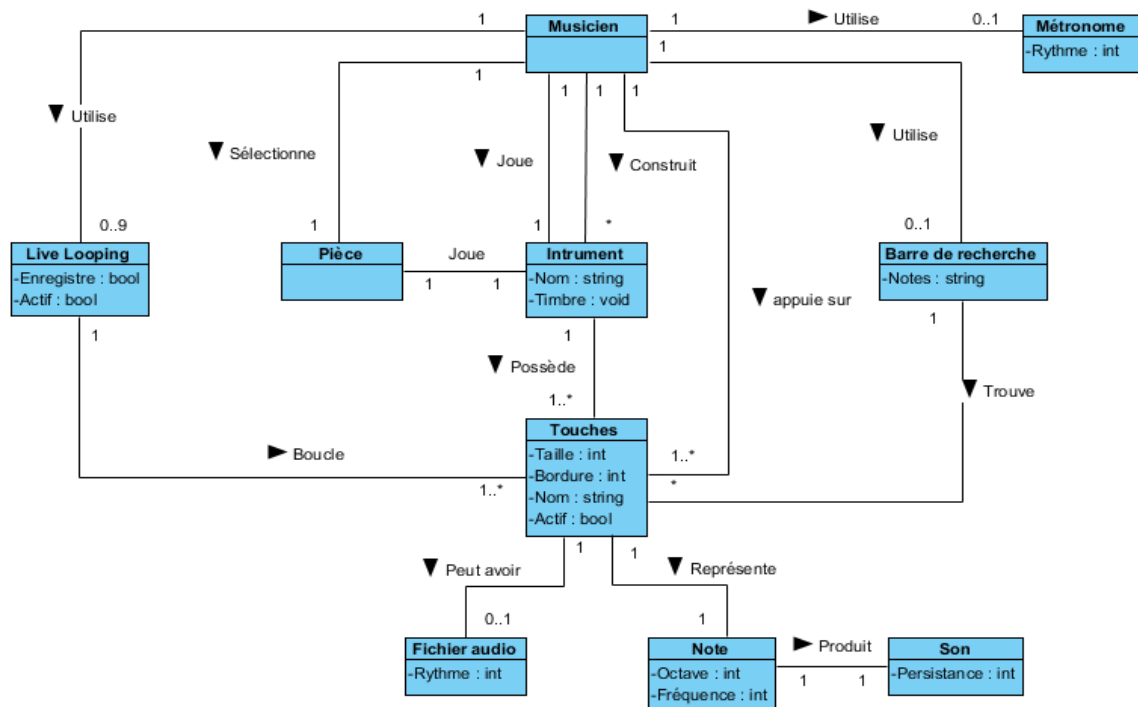
Section 1 - Énoncé de vision

L'équipe Marley est fier de présenter son tout dernier projet : Le Gaudrophone. Il s'agit d'une application développée pour la firme Gaudreault-Lanois ayant pour but de permettre la création d'instrument de musique et de rendre cette création accessible à un très grand public principalement par sa simplicité d'utilisation. Tout utilisateur sera en mesure de créer son propre instrument personnalisé et de le maîtriser à la perfection.

En plus de permettre la création de différents instruments, il est aussi possible de se pratiquer avec ceux-ci. L'utilisateur peut en premier lieu se pratiquer par lui même en jouant des pièces qu'il connaît. Il doit seulement avoir les notes nécessaire à la réalisation de sa pièce pour la pratiquer. Si l'instrument en question n'a pas toutes les notes requises, l'utilisateur peut toujours les créer lui-même. L'autre façon de se pratiquer est avec la lecture de pièce musicale. Cette option, en plus de jouer la pièce charger par l'utilisateur, affiche les notes jouées en temps réel. Le futur musicien peut donc apprendre les pièces avec une grande facilité et ainsi devenir encore plus performant avec son instrument. Un métronome sera toujours disponible pour les utilisateurs afin de se pratiquer à conserver le rythme musicale. Le bouclage en direct de séquences de notes permet aux musiciens expérimentés d'explorer leur côté créatif et ainsi de les aider à produire leur propre composition.

Le Gaudrophone est un projet innovateur dans le domaine musicale puisque celui-ci offre un accès immédiat à plusieurs instruments de musique. Il permet aussi à n'importe qui de devenir un excellent musicien grâce à ses options de pratique élaborées.

Section 2 - Modèle du domaine



2 - Diagramme de classe

L'utilisateur peut soit créer ou soit jouer d'un instrument. Il peut seulement créer ou jouer d'un instrument à la fois. Lorsqu'il joue d'un instrument, il peut activer le live-looping pour lui permettre d'enregistrer une série de notes et la faire rejouer en boucle. Cette fonction peut être activée jusqu'à 9 fois pour l'instrument joué. L'utilisateur peut également appuyer sur des touches pour les tester sans qu'elles fassent partie d'un instrument quelconque.

Les touches sont les principales composantes d'un instrument. Un instrument peut en posséder une ou plusieurs. Elles possèdent plusieurs caractéristiques. Certaines de ses caractéristiques sont intrinsèques et sont donc des attributs de cette classe conceptuelle. Cependant, les touches ont également des caractéristiques qui ne sont pas intrinsèques. Chaque touche possède une forme, elle peut soit posséder une couleur ou une image de fond et elle peut posséder ou non un fichier audio. L'utilisateur a également accès à une barre de recherche pour retrouver les touches qu'il désire jouer ou modifier.

L'utilisateur peut aussi inclure un fichier texte qui contient une partition. Cette partition est ensuite contrôlée par une barre de positionnement qui est elle-même contrôlée par l'utilisateur.

L'utilisateur peut également activer ou non un métronome au rythme de son choix pour l'aider lors de ses répétitions pour pratiquer son instrument.

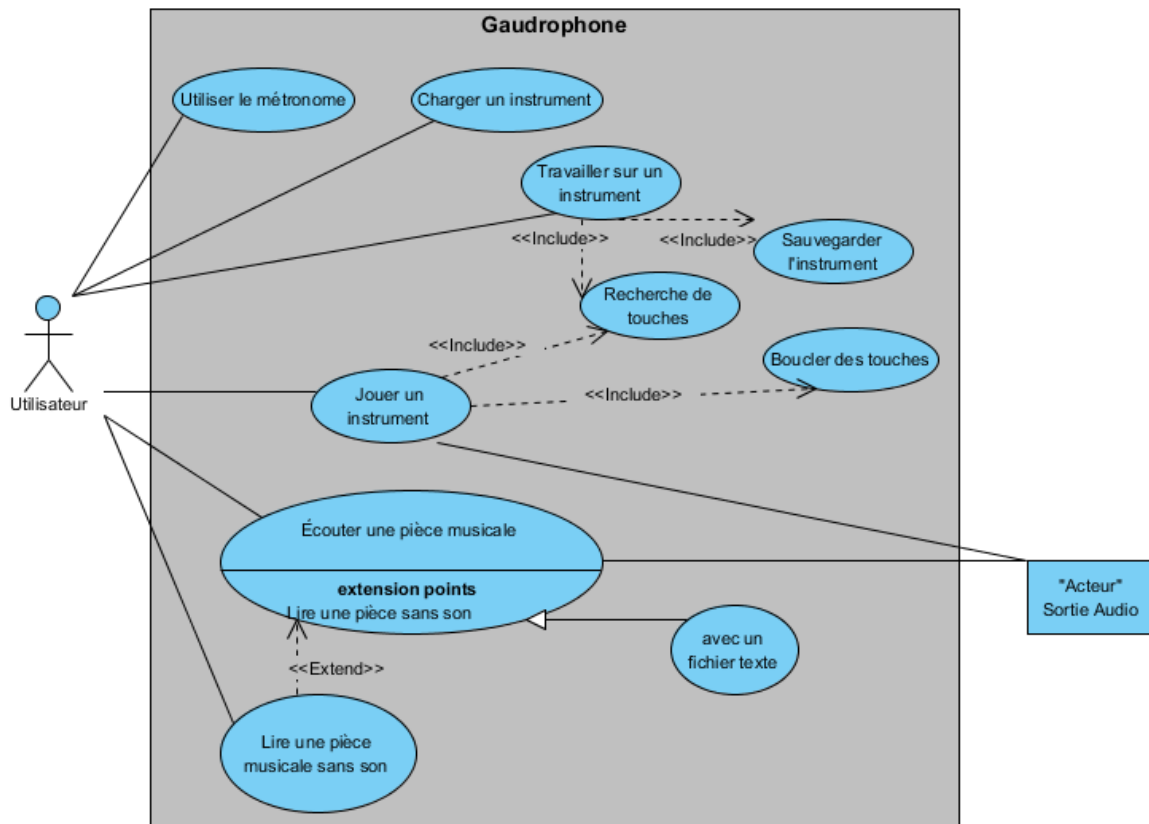
Section 3 - Glossaire

- **Artéfact** : Tout produit issu du travail effectué.
- **Fréquence** : Nombre de fois où une action se produit dans un temps donnée (1 seconde lorsque l'on parle de Hertz).
- **Octave** : Intervalle entre deux sons dont le ratio de leur fréquences fondamentales est de 2. Par exemple, si l'on prend la note standardisée de La de 440 Hz, l'octave précédente est de 220 Hz et l'octave suivante est de 880 Hz.
- **Persistance** : Fait de durer.
- **Timbre** : Qualité particulière du son, indépendante de sa hauteur ou de son intensité mais spécifique de l'instrument, de la voix qui l'émet. Il est lié aux intensités relatives des harmoniques qui composent le son.

Section 4 - Modèle des cas d'utilisation

Diagramme des cas d'utilisation

L'équipe a pris le temps d'analyser les besoins du client pour la réalisation du Gaudrophone afin de ressortir ce qui était nécessaire à notre projet pour réaliser l'application la plus complète possible. Le schéma ci-dessous est un diagramme des cas d'utilisation de notre application qui permet de placer en contexte le projet qu'est le Gaudrophone.



4-1 Diagramme des cas d'utilisation

Les tableaux qui suivent seront une description des différents cas d'utilisation de notre application. En fonction de leur importance, certains ont plus d'information que d'autres.

Texte des cas d'utilisation

Cas d'utilisation:	Créer une touche (inclus dans : Travailler sur un instrument)
Système :	Gaudrophone
Acteur(s) :	Utilisateur
Parties prenants et intérêts :	Utilisateur : Il désire ajouter touche à son instrument de musique.
Préconditions :	L'utilisateur travaille sur un instrument.
Garanties en cas de succès :	La nouvelle touche apparaît sur la page de l'instrument et elle est maintenant utilisable.

Scénario principale :	<p>1. L'utilisateur ouvre un projet</p> <p>2. L'utilisateur choisit de créer une nouvelle touche</p> <p>3. Le menu de création de touche apparaît</p> <p>4. L'utilisateur choisit l'apparence de la touche</p> <p>5. L'utilisateur choisit les caractéristiques auditives de la touche</p> <p>6. La touche est ajoutée à l'instrument</p>
Scénarios alternatifs	<p>4. L'utilisateur choisit l'apparence de la touche :</p> <p>1. Il choisit la forme désirée</p> <p>2. Il choisit la bordure désirée</p> <p>3a. Il choisit la couleur de fond</p> <p>3b. Il importe une image qui sera placée comme fond</p> <p>4. Il choisit les informations il veut voir sur la touche</p> <p>5. L'utilisateur choisit les caractéristiques auditives de la touche</p> <p>1a. Il choisit le nom de la note</p> <p>2a. Il choisit l'octave</p> <p>3a. Il choisit la fréquence</p> <p>4a. Il choisit la persistance</p> <p>1b. Il importe un fichier audio</p>

Cas d'utilisation:	Jouer un instrument
Système :	Gaudrophone
Acteur(s) :	Utilisateur, Sortie audio
Parties prenantes et intérêts :	Utilisateur : Il désire jouer d'un instrument. Sortie audio : Il produit le son de l'instrument et le rend audible pour l'utilisateur.
Préconditions :	L'instrument est existant et contient au moins une touche.
Garanties en cas de succès :	Les touches produisent des sons lorsqu'elles sont activées. Une magnifique mélodie est entendue par les gens aux alentours grâce à la sortie audio.

Scénario principale :	1. L'utilisateur charge un instrument	2. Le système va chercher l'instrument en mémoire. L'instrument apparaît
	3.L'utilisateur enfonce une touche	4. Le son demandé est envoyé dans la sortie audio
		5. La touche prend une apparence plus dominante face aux autres touches de l'instrument.
		7. Le son continue d'être envoyé
	6. L'utilisateur maintient la touche	9. Le son arrête de jouer
	8. L'utilisateur relâche la touche	
	Les étapes 3-9 se réalisent pour chaque touche de l'instrument	11. La ou les notes concernées prennent une apparence plus dominante
	10. L'utilisateur utilise la barre de recherche	
	12. L'utilisateur quitte la session musicale	

Cas d'utilisation:	Écouter une pièce musicale
Système :	Gaudrophone
Acteur(s) :	Utilisateur, Sortie audio
Parties prenantes et intérêts :	Utilisateur : Il désire entre une pièce musicale jouée par un instrument. Sortie audio : Elle génère le son de la pièce musicale.

Préconditions :	L'utilisateur a un instrument ouvert et celui-ci a toutes les notes que la pièce musicale contient. Le fichier qui contient la pièce musicale a comme format celui accepté par l'application.
Garanties en cas de succès :	La pièce est jouée par la sortie audio.
Scénario principale :	<p>1. L'utilisateur charge un instrument</p> <p>2. Le système va chercher l'instrument en mémoire. L'instrument apparaît</p> <p>3. L'utilisateur importe une pièce musicale</p> <p>4. Le système vérifie que la pièce peut être jouée</p> <p>5. La pièce apparaît selon un format d'affichage prédéfini</p> <p>6. Une barre de positionnement fait son apparition</p> <p>7. Les notes qui se trouvent dans la pièce changent de couleur</p> <p>8. La pièce commence et le son est envoyé à la sortie audio</p> <p>9. Les notes jouées par la pièce prennent une apparence plus dominante face aux autres touches de l'instrument.</p> <p>10. L'utilisateur pause la pièce à l'aide de la barre de positionnement</p> <p>11. Le son n'est plus envoyé, tout le reste de l'affichage ne bouge plus</p> <p>12. L'utilisateur avance ou recule dans la pièce à l'aide de la barre de positionnement</p> <p>13. L'affichage se déplace à la bonne place dans la pièce</p> <p>14. Les notes en apparence plus dominante changent pour les bonnes</p>

	<p>16. Le son est envoyé à la sortie audio</p> <p>15. L'utilisateur fait repartir la pièce</p> <p>17. L'affichage supplémentaire disparaissent</p> <p>La pièce se termine</p>
--	--

Cas d'utilisation:	Boucler des touches
Système :	Gaudrophone
Acteur(s) :	Utilisateur, Sortie audio
Parties prenantes et intérêts :	Utilisateur : Il désire faire jouer une série de son en boucle Sortie audio : Il produit le son des touches en boucle
Préconditions :	L'instrument est existant et contient au moins une touche.
Garanties en cas de succès :	Les sons bouclés sont joués jusqu'à temps que l'utilisateur arrête le bouclage.
Scénario principale :	<p>1. L'utilisateur charge un instrument</p> <p>2. Le système va chercher l'instrument en mémoire. L'instrument apparaît</p> <p>3. L'utilisateur appuie sur une touche de bouclage</p> <p>4. Le système commence l'enregistrement de la boucle</p> <p>5. L'utilisateur joue quelques notes</p> <p>6. L'utilisateur appuie sur la touche de bouclage</p> <p>7. Le système arrête l'enregistrement</p> <p>8. L'enregistrement est envoyé dans la sortie audio et se répète en boucle</p>

	<p>9. L'utilisateur appuie pour une troisième fois sur la touche de bouclage</p> <p>10. L'enregistrement cesse d'être joué par la sortie audio</p> <p>11. Le système supprime l'enregistrement</p>
--	--

Cas d'utilisation:	Travailler sur un instrument
Système :	Gaudrophone
Acteur(s) :	Utilisateur
Parties prenants et intérêts :	Utilisateur : Il désire créer un tout nouvel instrument ou tout simplement continuer de travailler sur un ancien instrument
Préconditions :	L'instrument a été sauvegardé auparavant si l'utilisateur veut travailler sur celui-ci
Garanties en cas de succès :	L'utilisateur sera en mesure d'utiliser son instrument pour jouer de celui-ci
Scénario principale :	<p>1. L'utilisateur ouvre l'application</p> <p>2. L'utilisateur crée une nouvelle touche</p> <p>3. Le système place la touche sur la page de l'instrument</p> <p>Les étapes 2-3 sont répétés tant que l'utilisateur ne veut plus de nouvelles touches</p> <p>4. L'utilisateur utilise la barre de recherche pour vérifier s'il a bel et bien créé une certaine note</p> <p>5. La ou les notes concernées prennent une apparence plus dominante</p> <p>6. L'utilisateur sauvegarde son nouvel instrument</p> <p>7. Le système enregistre en mémoire l'instrument</p>

	8. L'utilisateur ferme l'application
Scénario alternatif	<p>1. L'utilisateur ouvre l'application</p> <p>2. L'utilisateur charge un ancien instrument</p> <p>3. Le système va chercher l'instrument en mémoire. L'instrument apparaît</p> <p>4. L'utilisateur modifie une touche</p> <p>5. Les modifications de la touche sont appliquées</p> <p>6. L'utilisateur sauvegarde son nouvel instrument</p> <p>7. Le système enregistre en mémoire l'instrument</p> <p>8. L'utilisateur ferme l'application</p>

Cas d'utilisation:	Recherche de touches
Acteur(s) :	Utilisateur
Type :	Secondaire
Description :	L'utilisateur utilise une barre de recherche pour trouver les touches correspondantes à ce qui est inscrit dans la barre de recherche. Les touches concernées prennent une apparence plus dominante que le reste.

Cas d'utilisation:	Sauvegarder l'instrument
Acteur(s) :	Utilisateur
Type :	Secondaire
Description :	Suite à la réalisation de son instrument, l'utilisateur utilise la sauvegarde pour conserver son instrument pour les prochaines utilisations de l'application. L'instrument est conservé en mémoire.

Cas d'utilisation:	Charger un instrument
Acteur(s) :	Utilisateur
Type :	Primaire
Description :	L'utilisateur va chercher en mémoire un ancien instrument qu'il a précédemment sauvegardé. Celui-ci s'affiche sur la page de l'application.

Cas d'utilisation:	Lire une pièce musicale sans son
Acteur(s) :	Utilisateur, Sortie audio
Type :	Primaire
Description :	L'utilisateur charge une pièce musicale ainsi qu'un instrument. L'application charge la pièce musicale. Celle-ci est jouée sans produire aucun son. Lorsque les notes qui sont jouées dans la pièce elles prennent une apparence plus dominante. Si l'utilisateur joue les notes dominantes, ceux-ci produisent le son attendu.

Cas d'utilisation:	Modifier une touche (inclus dans : Travailler sur un instrument)
Acteur(s) :	Utilisateur
Type :	Secondaire
Description :	Lorsque l'utilisateur travaille sur un instrument, il peut choisir une touche parmi celle déjà existante et la modifier. Il peut alors changer l'apparence et les caractéristiques auditives de celle-ci. Les modifications s'appliquent directement sur la touche,

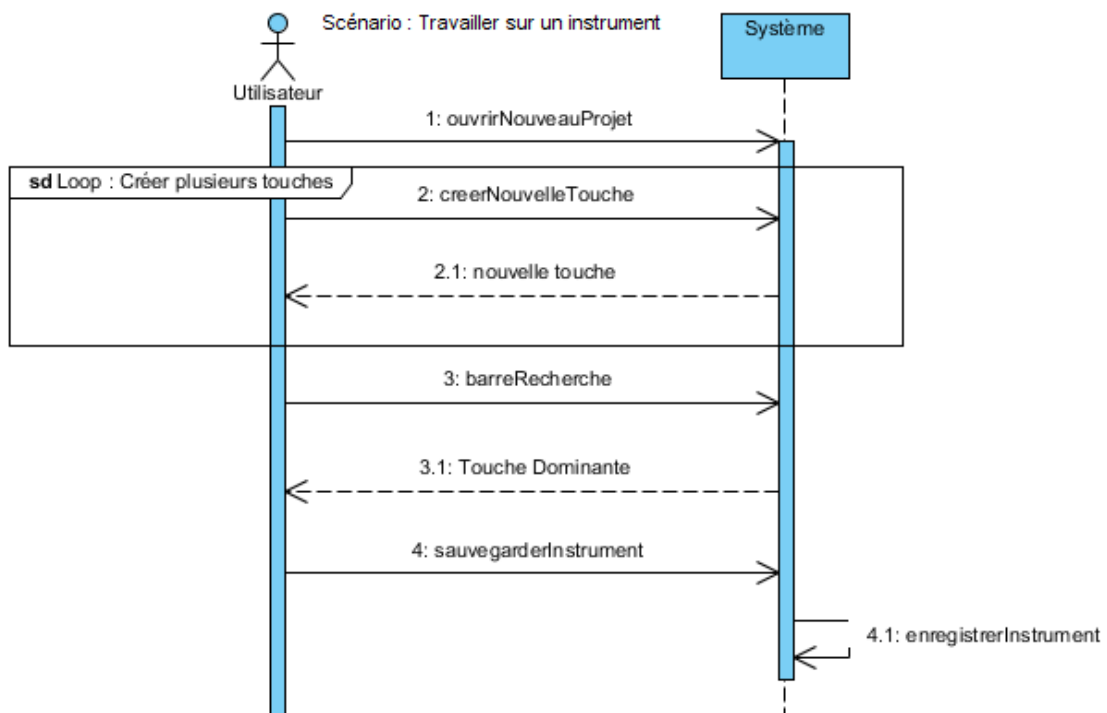
Cas d'utilisation:	Utiliser le métronome
Acteur(s) :	Utilisateur, Sortie audio
Type :	Secondaire

Description :	Lorsque l'utilisateur démarre le métronome, un bruit qui se répète à une certaine vitesse et qui ressemble à un tic d'horloge est entendu par l'utilisateur grâce à la sortie audio qui envoie le bruit. La fréquence du métronome peut être modifiée par l'utilisateur.
---------------	--

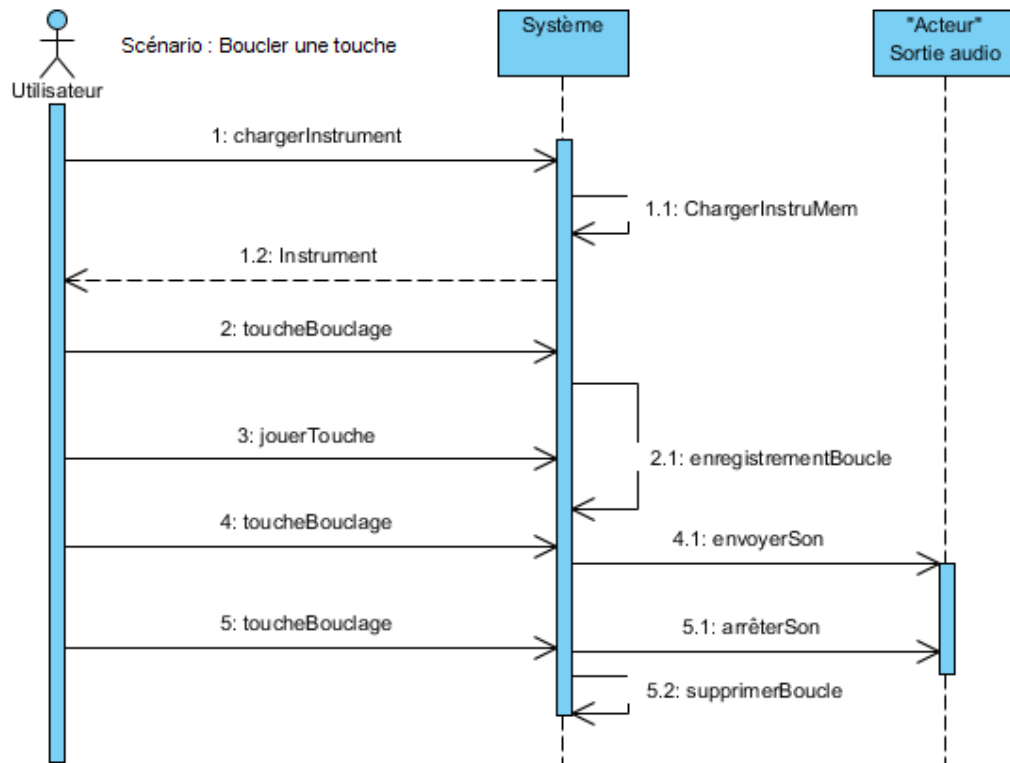
Diagramme de séquence système

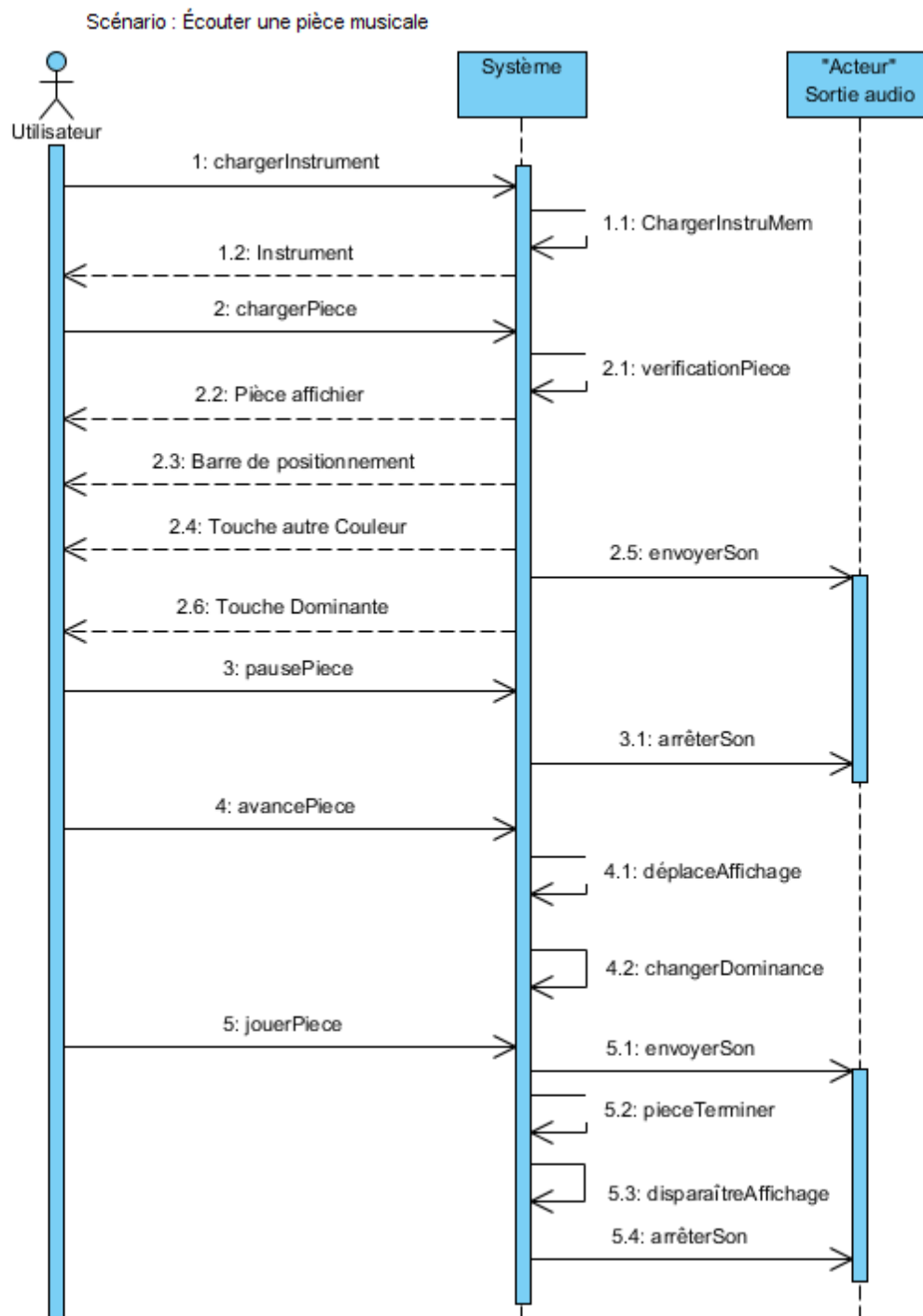
Les diagrammes qui suivent sont des diagrammes de séquence système. Ils ont été produits pour les cas d'utilisations les plus importants, soit les cas les plus détaillés de la section précédente.

4-2 scénario :Travailler sur un instrument

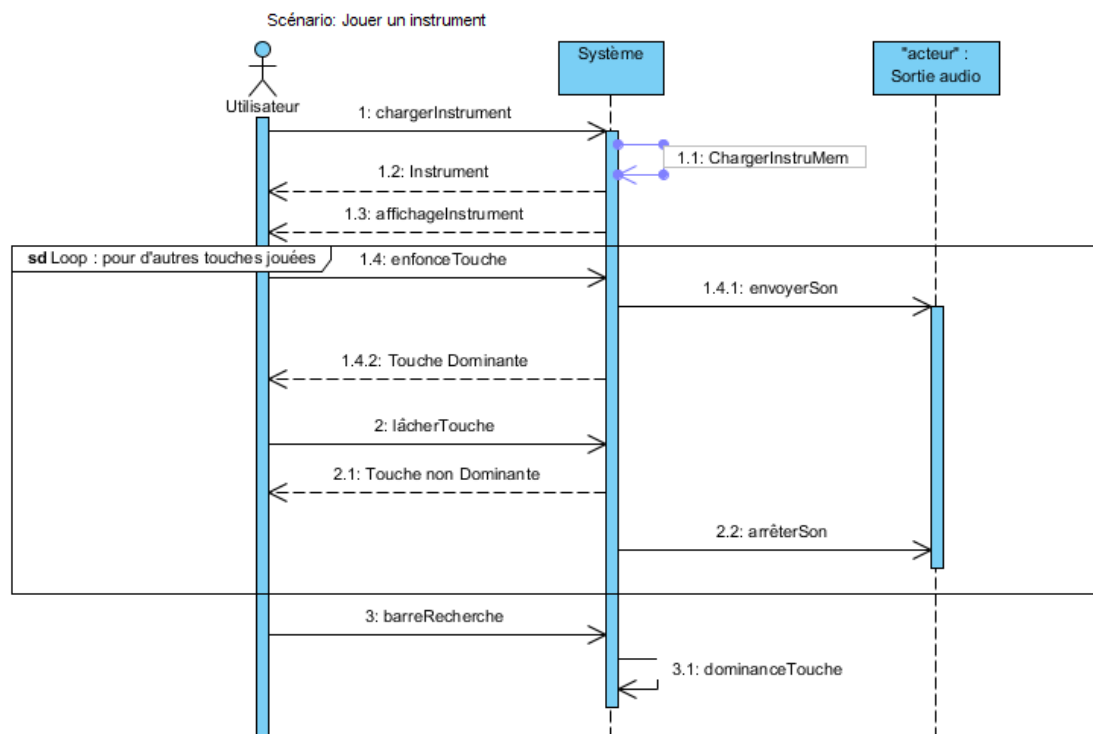


4-3 scénario :Boucler une touche

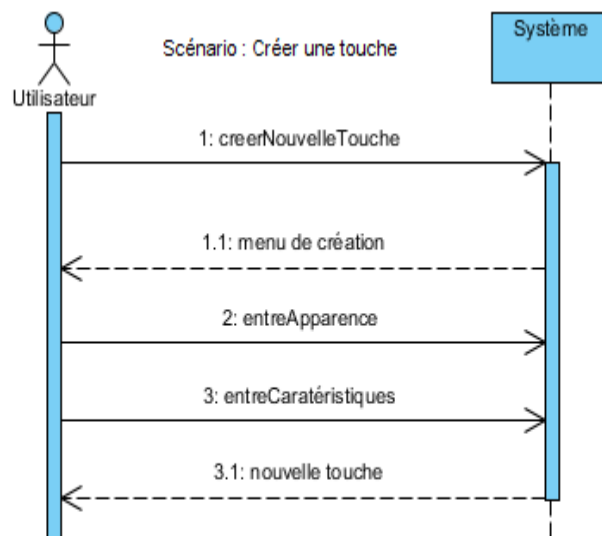




4-4 scénario : Écouter une pièce musicale





4-5 scénario :Jouer un instrument

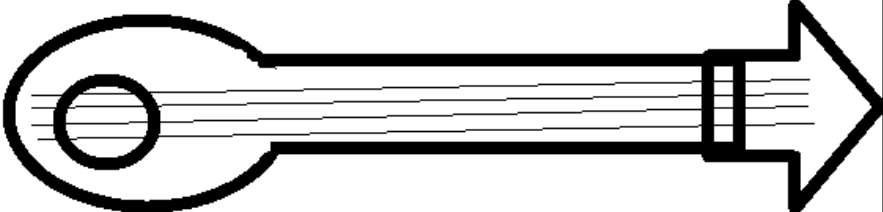









4-6 scénario :Créer une touche

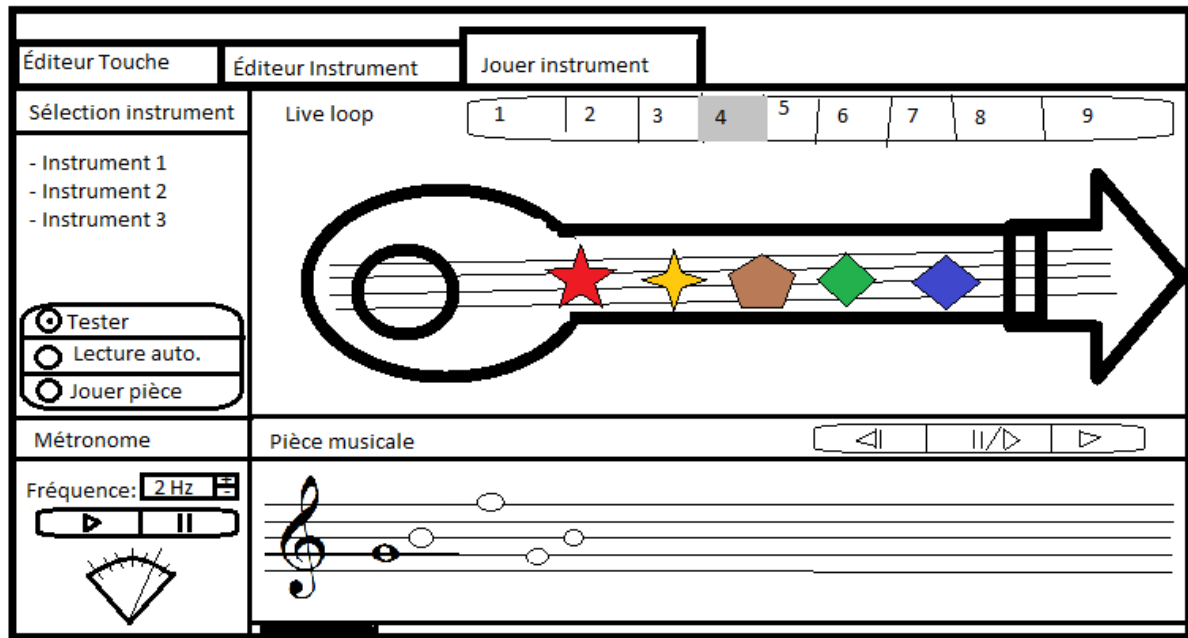
Section 5 - Esquisses des interfaces utilisateur

Éditeur Touche		Éditeur Instrument	Jouer instrument
Sélection touche			
<div>Rechercher </div> <ul style="list-style-type: none">- Touche 1- Touche 2- Touche 3			
<div>Nouveau</div> <div>Supprimer</div> <div>Enregistrer</div>		Apparence	Caractéristiques
		<div>- Taille : <input type="text"/></div> <div>- Forme : <input type="text"/></div> <div>- Couleur : <input type="text"/></div> <div>- Bordure : <input type="text"/></div> <div>- Image de fond : <input type="text"/></div>	<div>- Nom de la note : <input type="text"/></div> <div>- Octave : <input type="text"/></div> <div>- Fréquence : <input type="text"/></div> <div>- Persistance : <input type="text"/></div> <div>- Fichier audio : <input type="text"/></div>

5-1 Esquisse : Éditeur de touche

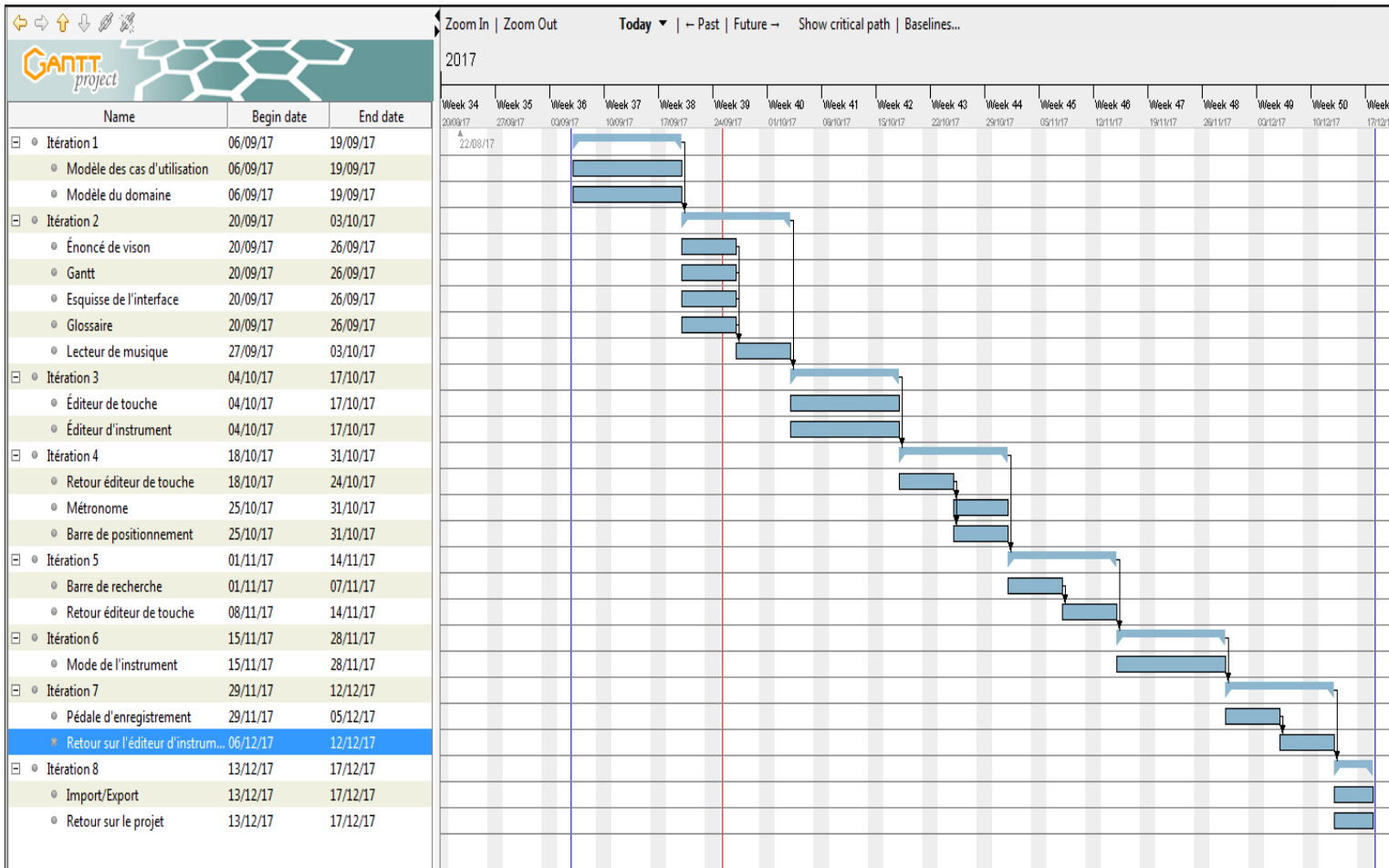
Éditeur Touche		Éditeur Instrument	Jouer instrument
Sélection instrument			
<ul style="list-style-type: none">- Instrument 1- Instrument 2- Instrument 3 <div>Nouveau</div> <div>Supprimer</div> <div>Enregistrer</div>			
Détails	Notes <div>Rechercher </div>		
<div>- Nom: <input type="text"/></div> <div>- Timbre: <input type="text"/></div> <div>- Fichier pièce: <input type="text"/></div>	<div>      ...</div> <div>Note_1 Note_2 Note_3 Note_4 Note_5</div>		

5-2 Esquisse : Éditeur d'instrument



5-3 Esquisse :Jouer d'un instrument

Section 6 - Diagramme de Gantt



6 - Diagramme de Gantt