Composer et interpréter de la musique interactive en ultilsant le langage HipHop.js

Petit Bertrand

22

Composer et interpréter de la musique interactive en utilisant le langage HipHop.js

Bertrand Petit

Institut national de recherche en sciences et technologies du numérique/
Centre international de rencontres mathématiques (Inria/Cirm)
Sophia Antipolis, Nice

DOI → https://doi.org/10.25836/rnrm.2020.46

SKINI

Skini est une plateforme de composition et de production de spectacles vivants permettant la participation du public à l'aide d'appareils connectés (smartphones, tablettes, PC, etc.). Le compositeur crée au préalable des éléments de base : des patterns – ou schémas – mélodiques ou sonores, des instruments, des groupes d'instruments et une partition dynamique qui régit la manière dont ces éléments de base vont se comporter en fonction d'événements produits par le public. La plateforme a pour objectif de contrôler la qualité musicale de l'œuvre lorsque, durant le concert ou la performance, le public interagit avec le système et donne naissance à une création musicale originale. Les « partitions » Skini sont exprimées selon des contraintes sur des événements qui contrôlent quels éléments musicaux sont accessibles au public et quand ils sont disponibles. Les contraintes peuvent être instantanées – par exemple désactiver les violons pendant que les trompettes jouent. Elles peuvent également être temporelles – par exemple empêcher le piano de jouer plus de 30 secondes consécutives. La plateforme Skini est mise en œuvre avec Hop.js (Serrano et Prunet 2016) pour l'infrastructure générale et la plupart des interfaces utilisateur, et avec HipHop.js (Berry et Serrano 2020) qui est le langage utilisé pour les partitions. Les constructions du langage HipHop.js consistent en des opérateurs temporels tels que des exécutions parallèles, des séquences, des attentes, des points de synchronisation et des préemptions. Elles constituent le langage de base pour l'expression des contraintes musicales de Skini. Cet article présente les principes de la plateforme Skini. Quelques pièces musicales créées avec Skini se trouvent à l'adresse suivante : https://soundcloud.com/user-651713160

INTRODUCTION

Dans les années 60, le philosophe Umberto Eco et des musiciens tels que K. Stockhausen, K. Penderecki et L. Berio se sont interrogés sur les relations entre les compositeurs, les musiciens et le public (Eco 1965). Eco a montré qu'à la suite de l'évolution de la physique de Copernic à Einstein, la perception du monde est passée d'une perception statique à une perception dynamique. Cela a eu un impact sur l'art du xxe siècle où certains musiciens ont essayé d'exprimer cette dynamique à travers des œuvres où l'interprète et le public avaient un impact concret sur le résultat musical. Eco a qualifié ces tentatives d'œuvres « ouvertes » ou d'œuvres « en mouvement », dont la production musicale finale n'est pas strictement connue à l'avance. Les récentes améliorations technologiques, principalement la large couverture des réseaux de télécommunications et la disponibilité des appareils mobiles, ont ouvert de nouvelles possibilités pour

la musique interactive qui est devenue un domaine actif (Arango et Giraldo, 2016 ; Lee, Essl, et Mao, 2014 ; Weitzner, et al., 2012). Un important effort a été consacré aux développements techniques des systèmes permettant des interactions individuelles, SWARMED (Hindle, 2013) en est un exemple. Ce sont ces défis technologiques qui ont le plus attiré l'attention, de sorte que les outils consacrés à la composition musicale de performances interactives proposant un schéma de composition clair sont rares. Skini est un de ces outils.

CONCEPTS DE BASE

Skini organise la musique à partir de courtes séquences musicales appelées patterns. Jouer la musique consiste à sélectionner et jouer certains de ces patterns. C'est le public pendant le spectacle qui doit sélectionner des patterns parmi un ensemble de disponibilités qui dépendent des sélections précédentes du public et du parcours musical décrit dans la partition du compositeur. Afin d'aider les participants à sélectionner les patterns, Skini permet d'abord de les écouter à l'aide d'un smartphone. Ils pourront ensuite être activés pour être joués par un système sonore (*digital audio workstation*) ou des musiciens. Pour garantir la musicalité de l'œuvre, le compositeur contrôle les patterns proposés au public en créant des groupes cohérents. L'art de gérer la disponibilité des groupes de patterns est l'orchestration, en référence à l'utilisation courante de ce mot en musique.

PATTERNS

Un *pattern* peut être n'importe quel élément d'une partition, une séquence MIDI ou un son préenregistré. Dans un souci d'interaction, les patterns sont généralement courts (quelques secondes), car, trop longs, ils monopoliseraient les instruments et limiteraient ainsi l'interaction avec le public. Lorsqu'il est activé, un *pattern* peut être joué tel quel ou peut être modifié en temps réel selon la façon dont le compositeur imagine le déroulement du spectacle. Par exemple, un *pattern* de séquence MIDI peut être transposé dans une autre tonalité. Il est aussi possible de permettre au public de créer des patterns à l'aide d'un outil de conception dédié. Comme les patterns sont activés par le public, le compositeur ne peut pas contrôler à l'avance le résultat de l'interprétation de sa partition. Il doit gérer l'incertitude introduite par l'interaction en pensant à l'œuvre en termes d'architecture de groupes de patterns. C'est en gérant la disponibilité des groupes de patterns que se définit le projet esthétique. L'utilisation de patterns pour créer une musique originale n'est pas nouvelle. Des compositeurs du

xx[e] siècle tels que Stockhausen (*Klavierstück* XI), Henri Pousseur (*Scambi*) et Boulez (*Troisième Sonate pour piano*) ont utilisé des concepts similaires. L'originalité de Skini est la délégation au public de l'activation des patterns. Comme ce ne sont plus les musiciens sur scène qui jouent la musique, l'artiste est obligé d'abandonner la manière traditionnelle de composer. L'interaction avec le public doit être inhérente au processus de créativité. Cela donne une nouvelle dimension à l'ouverture de l'œuvre au sens d'Umberto Eco.

ORCHESTRATION

Dans le langage musical commun, l'orchestration est la description de l'entrée temporelle des instruments effectivement joués. Les instruments sont disposés en groupes : violons, altos, violoncelles, contrebasses, cors, percussions, etc. Le compositeur choisit les groupes selon le déroulement de son œuvre. Dans Skini, l'orchestration consiste à grouper des patterns et définir des contraintes sur l'entrée de ces groupes. L'orchestration régit l'interaction avec le public et celui-ci réagit à ce qu'il écoute. Le comportement du public est donc contraint par les combinaisons de patterns disponibles à un instant. Un tel système, réagissant aux événements (activation et temps) pour atteindre un état défini de l'orchestration, est modélisé par des « automates ». Le langage HipHop.js a été précisément créé pour la programmation des automates. Il peut mettre en œuvre des stratégies complexes fondées sur le comportement du public ou des stratégies simples en déléguant la majeure partie du contrôle à un contrôleur qui supervise manuellement le spectacle.

INTERACTION

Skini propose *trois* rôles différents pour les personnes dans l'audience :

– Acteurs : ce sont ceux qui activent des patterns en utilisant le navigateur web de leur smartphone ou tablette (**figure 1 [i]**). En sélectionnant des patterns, les acteurs génèrent des événements qui sont traités par l'automate d'orchestration.

– Concepteurs : ils créent des patterns pendant le spectacle à l'aide d'un séquenceur. Celui-ci fonctionne dans un navigateur web (**figure 1[ii]**). Il est accessible par des smartphones ou des tablettes et permet de créer des boucles synchronisées avec tous les autres patterns.

– Conducteurs : sont ceux qui donnent leur avis ou proposent une évolution globale de la performance. Leur interface web n'est pas présentée ici.

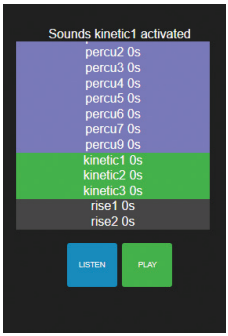

Fig. 1 [i]. Interface acteur.

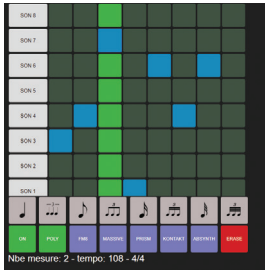

Fig. 1 [ii]. Le séquenceur Skini.

PROCESSUS DE COMPOSITION

Voici les étapes que suit le compositeur en utilisant Skini : 1. conception de tous les patterns ; 2. création de l'orchestration initiale ; 3. simulation du comportement de l'audience ; 4. modification de l'orchestration et répétition de l'étape 3 jusqu'à ce que le résultat soit conforme au projet esthétique.

CONCEPTION DES PATTERNS

Un pattern peut être un fragment d'une phrase mélodique, un son de synthèse complexe, ou un échantillon sonore, etc. Les figures 2, 3 et 4 montrent trois représentations d'un même pattern simple qui peut être joué sous trois formes différentes : La figure 2 par un synthétiseur, la figure 3 par un musicien, la figure 4 par un lecteur de fichiers son.

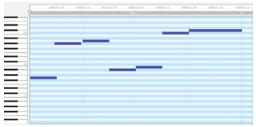

Fig. 2. Exemple de pattern MIDI simple.

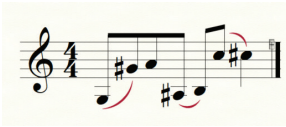

Fig. 3. Exemple d'un score de modèle simple.

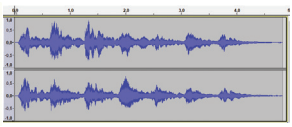

Fig. 4. Exemple d'un pattern sonore.

La principale difficulté pour le compositeur est d'imaginer des patterns adaptés à tous les scénarios orchestraux – ou à la plupart d'entre eux – qui seront créés lors de la deuxième étape.

LA CONCEPTION D'ORCHESTRATION

Le compositeur conçoit l'évolution d'une orchestration qui doit anticiper les interactions du public. Pour cela, l'orchestration doit s'adapter à des parcours variés selon les événements générés par le public. Cette étape consiste à programmer avec HipHop.js qui est un langage de haut-niveau permettant de décrire d'une façon intuitive le fonctionnement de ce qui, en informatique, est nommé « automate ». La programmation des automates est un sujet complexe auquel HipHop.js apporte une solution élégante, souple et solide.

SIMULATION

Il est difficile d'imaginer comment la mise en œuvre de l'orchestration va réellement fonctionner, et c'est essentiellement l'art de composer de la musique avec Skini qui permet de faire face à cette incertitude et à cette ouverture. C'est pourquoi les patterns et l'orchestration ont besoin d'outils d'une simulation du résultat final. Skini propose un simulateur qui se comporte comme un public qui choisirait des patterns au hasard. Le simulateur est contrôlé par trois paramètres : les temps de réponse minimum et maximum des personnes dans le public et le temps d'attente pour qu'un pattern soit joué. Bien que simples, ces trois paramètres sont suffisants pour simuler des performances réalistes et permettre au compositeur d'avoir une bonne idée de la forme que prendra de la musique qu'il compose.

CONCLUSION

Skini est une plateforme de compositions et de réalisation de performances interactives. Elle prend en compte le comportement du public. Contrairement à la musique où les partitions sont utilisées pour définir le discours musical en fonction du temps, avec Skini le compositeur doit réfléchir au comportement du public, au type d'interactions qu'il souhaite en fonction de l'esthétique de son projet. La plateforme propose une solution qui repose sur quelques concepts de base et dont la mise en œuvre musicale est rendue possible grâce à un langage informatique particulier, HipHop.js. L'exécution de différentes pièces de musique utilisant Skini, ont permis de démontrer que nos concepts de base permettent de réaliser des performances où les auditeurs se sentent impliqués dans la production d'un ensemble musical cohérent. En outre, l'utilisation de Skini lors de projets pédagogiques est prometteuse. Skini donne la possibilité de progresser pas à pas, en groupe et en interaction avec un compositeur ou un professeur tout en suivant une méthodologie simple. Skini a été utilisé pour un

projet SACEM (Fabrique à musique 2018) avec des collégiens. Ceci a permis d'aborder divers sujets musicaux allant de la conception de phrases musicales simples à la conception d'un discours musical complet, afin de produire une véritable performance musicale.

BIBLIOGRAPHIE

Arango, Julián, Julian Jaramillo et Daniel Melán Giraldo. « The Smartphone Ensemble. Exploring Mobile Computer Mediation in Collaborative Musical Performance », *Proceedings of the International Conference on New Interfaces for Musical Expression*, vol. 16, 2016, p. 61-64.

Berry, Gérard et Manuel Serrano. « HipHop.js: (A) Synchronous Reactive Web Programming», *PLDI 2020: Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation, London*, New York, Association for Computing Machinery, p. 533-545, 2020.

Eco, Umberto. *L'Œuvre ouverte*, Paris, Points, 1965.

Hindle, Abram. « SWARMED: Captive portals, Mobile Devices, and audience Participation in Multi-User Music Performance », *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2013, p. 174-179.

Lee, Sang Won, Georg Essl, et Zhuoqing Morley Mao. « Distributing Mobile Music Applications for Audience Participation Using Mobile Ad-hoc Network (MANET) », *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2014, p. 533-536.

Serrano, Manuel et Vincent Prunet. « A Glimpse of Hopjs », *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming*, New York, ACM, 2016.

Weitzner, Nathan, Jason Freeman, Stephen Garrett et Yan-ling Chen. « massMobile – an Audience Participation Framework », *NIME 2012 Proceedings of the International Conference on New Interfaces for Musical Expression*, Ann Arbor, University of Michigan, 2012, p. 92-95.