CRÉATION MUSICALE COLLABORATIVE EN TEMPS RÉEL SUR LE WEB : EXPLOITATION DES WEB AUDIO MODULES POUR LA PERFORMANCE MUSICALE ET LA COMPOSITION INTERACTIVES

Michel Buffa
Université Côte d'Azur,
CNRS, INRIA
France

Laurent Pottier
Université Jean Monnet,
laboratoire ECLLA
Saint-Etienne, France

RÉSUMÉ

Les technologies web permettent aujourd'hui le développement d'environnements de création musicale collaborative accessibles depuis un navigateur. Cet article présente *sequencer.party*, une plateforme de création et de performance collaborative fondée sur des Web Audio Modules (WAM) interconnectés au sein de graphes modulaires partagés. La synchronisation repose sur un algorithme de type CRDT garantissant la cohérence des sessions sans transmission de flux audio. Nous décrivons l'architecture du système, ses fonctionnalités collaboratives et son utilisation en performance et en pédagogie. Une expérimentation avec des étudiants montre que cet environnement favorise l'appropriation d'outils modulaires, la créativité distribuée et l'émergence de nouvelles pratiques, notamment l'usage de composants programmables et d'assistants d'intelligence artificielle. Ces résultats soulignent le potentiel des WAM et des technologies web pour la création musicale collaborative et l'enseignement.

1. INTRODUCTION

Les applications musicales collaboratives en temps réel sont de plus en plus répandues, permettant aux musiciens de co-créer de la musique indépendamment de leur situation géographique [8]. Grâce aux progrès des technologies web, notamment les APIs Web Audio, Web MIDI et le langage WebAssembly, le navigateur est devenu une plateforme viable pour ce type d'applications.

Les applications musicales collaboratives basées sur le Web peuvent être classées en plusieurs catégories. Les plateformes de performance musicale en réseau permettent à plusieurs utilisateurs de jouer simultanément via Internet, en utilisant des technologies telles que WebRTC ou WebSockets pour l'échange de données audio ou MIDI à faible latence [23]. Les stations de travail audio-numériques collaboratives (DAWs) basées sur le Web, telles que BandLab ou Audiotool, permettent la création et l'édition collective de projets musicaux directement dans le navigateur [4]. Les outils éducatifs, tels que Multi Web Audio Sequencer [14] et Soundcool Online [10], offrent des environnements collaboratifs adaptés à l'apprentissage et à l'expérimentation musicale, tandis que des *patchers* web tels que WebPD [11], Kiwi [20], JSPatcher [22] permettent de concevoir des systèmes musicaux modulaires inspirés d'environnements comme Pure Data ou Max/MSP. Dans le contexte web, la performance musicale collaborative peut reposer soit sur la transmission directe de flux audio entre participants, soit sur l'échange de données MIDI ou la synchronisation d'état des composants logiciels. Cette seconde approche, utilisée dans cet article, permet de réduire les contraintes de latence et de bande passante, tout en garantissant une cohérence structurelle de la performance.

Nous présentons ici une étude de cas centrée sur *sequencer.party*, une plateforme de création musicale collaborative développée pendant la pandémie de COVID-19 par Tom Burns[1], en collaboration avec l'équipe à l'origine du standard de plugins Web Audio Modules version 2 (WAM). *Sequencer.party* permet aux utilisateurs de créer et modifier collaborativement des installations musicales directement dans le navigateur, en assemblant des graphes de plugins WAM comprenant générateurs de notes, instruments, effets, modulateurs et composants visuels. Ces graphes constituent des installations musicales modulaires, comparables à des studios virtuels.

Cet article apporte trois contributions principales :

1. la conception et la validation d'une architecture de création musicale collaborative basée sur les Web Audio Modules et un mécanisme de synchronisation d'état fondé sur CRDT;
2. la démonstration de la viabilité de cette approche pour des performances musicales distribuées et hybrides, incluant l'intégration d'instruments matériels et de participants distants;
3. une étude de cas pédagogique analysant l'appropriation du système, la créativité distribuée et l'émergence de nouvelles pratiques, notamment l'intégration de programmation collaborative et d'assistance par intelligence artificielle.

La section 2 présente les caractéristiques des Web Audio Modules adaptées aux applications collaboratives. La section 3 décrit les fonctionnalités de *sequencer.party.* La section 4 détaille les mécanismes de synchronisation collaborative, tandis que la section 5

[1]Tom Burns est ingénieur logiciel et musicien électronique à Ottawa.

traite de la gestion des sessions en direct. La section 6 présente les performances hybrides combinant participants locaux et distants. La section 7 décrit une expérimentation menée avec des étudiants du Master Erasmus Mundus DIGICREA[2]. Enfin, les sections 8 et 9 présentent respectivement la conclusion et les perspectives.

2. LES WEB AUDIO MODULES

Le standard de plugins WAM fournit un cadre standardisé pour le développement de plugins audio haute performance pouvant fonctionner de manière transparente dans les navigateurs web, comparable conceptuellement aux plugins VST [2-3][6-7]. Les principales caractéristiques des WAMs sont *la modularité* (ce sont des composants audio réutilisables), *l'interopérabilité* qui garantit que les plugins peuvent fonctionner sur différentes applications hôtes – ce sont de purs *Composants Web*[3], *l'extensibilité* (prend en charge l'intégration avec diverses technologies web, notamment Web MIDI, WebAssembly, WebGL, WebGPU, ainsi qu'un ensemble d'extensions pour la 3D, la vidéo, la modulation des paramètres, la collaboration en temps réel, etc.), *la multimodalité* (ils peuvent être exécutés sans interface graphique, ou avec plusieurs interfaces graphiques, un WAM peut être l'hôte d'autres WAMs, etc.), et ce sont *des composants web de première classe :* ils ont tous une URI et peuvent être chargés et utilisés comme de simples ressources web - pensez aux images en HTML.

Des centaines de WAMs ont été développés par la communauté et sont accessibles via des points d'accès REST exposés par des serveurs de plug-ins distants. Ils peuvent être intégrés dans des applications hôtes à l'aide d'imports dynamiques JavaScript ou de manière déclarative en HTML, ce qui ne nécessite que très peu de lignes de code. L'écosystème WAM est vivant et en pleine évolution [6], et les contributions originales ouvrent de nouveaux paradigmes pour la collaboration musicale dans le navigateur. Un exemple en est l'utilisation récente des WAMs dans des applications musicales immersives 3D collaboratives [5].

Initialement proposés pour étendre les capacités de l'API Web Audio, le standard WAM est une interface standardisée pour le traitement en temps réel des signaux audio, MIDI et de contrôle. Comme nous le verrons, la conception des plugins WAMs les rend particulièrement adaptés aux applications collaboratives, en particulier celles qui nécessitent des interactions synchronisées entre plusieurs utilisateurs sur un réseau.

2.1. Entrées et sorties multimodales

En plus de gérer les flux audio en temps réel, les WAMs peuvent traiter les messages MIDI et OSC. Cette flexibilité est essentielle dans les environnements collaboratifs où les musiciens et les producteurs peuvent interagir simultanément avec des instruments virtuels, des surfaces de contrôle ou des processus algorithmiques. Par exemple, un synthétiseur WAM peut recevoir des notes MIDI d'un participant, des entrées audio d'un autre et des messages OSC d'un séquenceur ou d'une source de modulation, le tout traité en parallèle et en temps réel.

2.2. Synchronisation des états

Les WAMs exposent un ensemble complet de contrôles de paramètres via des getters et des setters standardisés. Ceux-ci sont essentiels pour mettre en œuvre l'automatisation, les liaisons UI et, plus important encore, le contrôle collaboratif des éléments partagés. L'API JavaScript fournit également les méthodes `getState()` et `setState()`, qui permettent la sérialisation et la désérialisation complètes de l'état du plugin. Il s'agit d'un mécanisme clé pour mettre en œuvre une *synchronisation fondée sur l'état*, une approche particulièrement adaptée aux applications web collaboratives, où une logique événementielle peut s'avérer inefficace ou excessivement consommatrice de bande passante. Contrairement aux architectures reposant sur la transmission immédiate de chaque interaction — susceptibles de provoquer des latences ou des problèmes de concurrence — ce modèle assure la cohérence en partageant ou en consolidant périodiquement l'état global d'un plugin ou de l'application.

Cette approche est compatible avec des techniques telles que l*es types de données répliqués sans conflit (CRDT)*, largement utilisées dans les éditeurs de texte collaboratifs.

3. ÉTUDE DE CAS : SEQUENCER.PARTY

Sequencer.party, un séquenceur musical collaboratif basé sur le Web constitue un outil très abouti démontrant le potentiel des Web Audio Modules dans des environnements collaboratifs. Cette application web met en évidence l'intégration de composants audio modulaires, la collaboration en temps réel et la programmation collaborative de certains composants par des participants connectés à distance, le tout directement dans le navigateur. Certains plug-ins WAM intégrés sont programmables de manière collaborative — par exemple des séquenceurs MIDI scriptables ou des modules intégrant des animations graphiques fondées sur des *ISF video shaders*[4] — et embarquent un éditeur de code dédié.

Nous verrons, dans la section 7 consacrée aux performances réalisées par des étudiants du master Erasmus Mundus DIGICREA, formés à la composition, aux techniques de studio, au traitement du signal et à l'informatique musicale, qu'avec l'appui d'assistants d'IA, ils sont parvenus, sans avoir besoin de

[2] https://master-digicrea.univ-st-etienne.fr/en/index.html

[3] https://developer.mozilla.org/en-US/docs/Web/API/Web_components

[4] https://editor.isf.video/about

connaissances préalables en programmation, à développer de nouveaux scripts de génération de notes ainsi que de nouveaux ISF video shaders, et à les intégrer à leur installation musicale dans *sequencer.party*, en vue d'une performance musicale.

3.1. Découverte de plugins et architecture modulaire

Sequencer.party intègre dynamiquement des plugins WAM en s'interfaçant avec le serveur de plugins de la communauté WAM[5]. Chaque plugin WAM exposé par ce service web (qui en propose plus d'une centaine) est décrit par un descripteur de métadonnées contenant des informations sur son type (par exemple, générateur de notes/contrôleur MIDI, instrument, effet, modulateur), ses entrées et sorties, et les catégories auxquelles il appartient. Ce processus de découverte permet aux utilisateurs d'intégrer de manière transparente un large éventail de plugins dans leur session, allant des synthétiseurs virtuels et des boîtes à rythmes aux effets audio et aux sources de modulation. Il est également possible d'ajouter un plugin en entrant son URI dans une fenêtre de dialogue, même s'il n'est pas proposé par défaut.

3.2. *Function Sequencer MIDI* collaboratif

Une fonctionnalité centrale de *sequencer.party* est l'intégration d'un *Function Sequencer* MIDI, un module programmable *collaborativement*, permettant de définir dynamiquement la génération et la transformation des événements MIDI. Selon le *preset* chargé, il peut se comporter comme un séquenceur pas à pas, un générateur algorithmique ou un transformateur MIDI (Figure 1, Figure 2). Chaque *preset* repose sur un programme JavaScript qui spécifie à la fois la logique musicale, les paramètres exposés et l'interface graphique (Figure 3).

Ce paradigme s'inscrit dans la continuité d'environnements programmables tels que Max/MSP ou les scripts JSFX de REAPER, transposé ici dans le Web sous la forme d'un Web Audio Module programmable et collaboratif.

Cette intégration étroite entre programmation, interaction musicale et collaboration étend les paradigmes existants de programmation musicale en y ajoutant une dimension collaborative native, permettant à plusieurs utilisateurs de co-concevoir et manipuler conjointement des instruments et processus génératifs au sein d'une même performance. Là encore, avec l'appui d'assistants d'IA, des étudiants du Master Erasmus Mundus DIGICREA, sans formation initiale en informatique, sont parvenus à s'approprier ces outils en intégrant des scripts générés à partir de simples requêtes textuelles et en fournissant aux assistants des exemples de code existants.

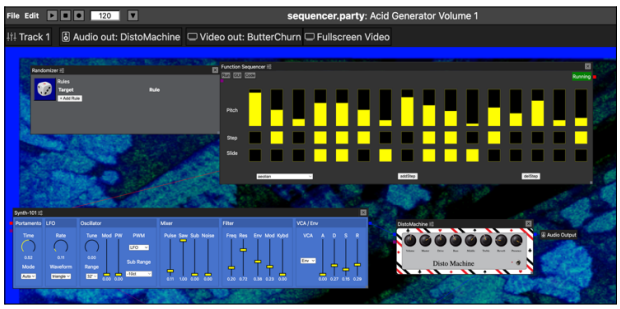

Figure 1 : WAM *Function Sequencer* programmable, ici chargé avec un *preset* de séquenceur pas à pas.

Figure 2 : *Function Sequencer* avec le *preset* Turing machine chargé - un puissant générateur de notes aléatoires.

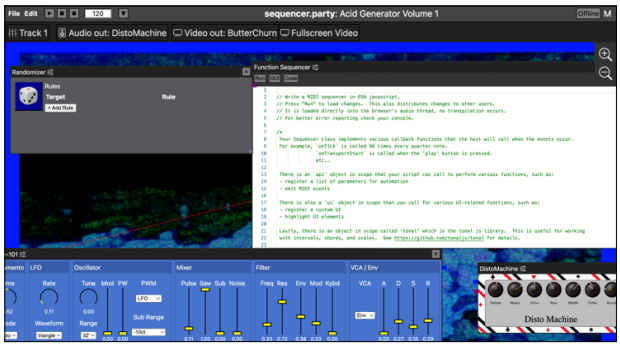

Figure 3 : à l'aide du bouton « code » de l'interface graphique du *Function Sequencer* de la Figure 1, le code source du *preset* actuel peut être édité en direct de manière collaborative.

3.3. Extensions WAM pour la collaboration et la gestion des ressources

La puissance et la flexibilité de sequencer.party sont encore renforcées par l'utilisation d'extensions WAM[6], telles que :

- **Extension "collaboration"** : permet aux plugins de gérer des documents texte édités de manière collaborative, garantissant un comportement cohérent dans les sessions d'édition partagées.
- **Extension "assets"** : permet aux plugins de charger ou d'enregistrer des ressources externes. Grâce à cette extension, l'environnement hôte peut gérer les opérations sur les fichiers (par exemple, le chargement/enregistrement de *presets* ou de code généré par l'utilisateur) sur son propre serveur,

[5] https://www.webaudiomodules.com/docs/community

[6] https://github.com/boourns/wam-extensions

facilitant ainsi le stockage persistant des *presets* du *Function Sequencer* et d'autres ressources de session.

Ces extensions jouent un rôle clé dans la mise en place de *workflows* collaboratifs viables et robustes dans un environnement basé sur un navigateur. Il existe plus de dix extensions WAM disponibles, voir [6] pour plus de détails.

3.4. Modulation des paramètres à la fréquence audio

Une fonctionnalité avancée de *sequencer.party* est la prise en charge de la modulation de paramètres à la fréquence audio. Certains plugins WAM peuvent agir comme des modulateurs et influencer dynamiquement les paramètres d'autres plugins. Lorsqu'un modulateur est connecté à un plugin cible, l'hôte expose automatiquement les paramètres disponibles, permettant leur automatisation en temps réel. Ce mécanisme permet de produire des effets expressifs tels que des balayages de filtres, des modulations d'amplitude ou des évolutions complexes de timbre, au sein d'un environnement modulaire.

Plusieurs types de modulateurs sont disponibles, notamment des modulateurs pas à pas, des générateurs d'enveloppe, des modulateurs aléatoires et des modulateurs basés sur des orbiteurs. La figure 4 illustre un modulateur pas à pas connecté à un synthétiseur WAM, permettant de faire évoluer certains paramètres dans le temps. La figure 5 montre un modulateur basé sur des *orbiteurs*, dans lequel des trajectoires bidimensionnelles contrôlent dynamiquement les paramètres du synthétiseur. Ces mécanismes reposent sur le système d'automatisation des paramètres intégré aux WAMs, garantissant une modulation précise et synchronisée.

3.5. Fonctionnalités supplémentaires

Au-delà des fonctionnalités de séquençage génératif et de routage modulaire, *sequencer.party* intègre un éditeur MIDI de type piano roll pour l'édition conventionnelle des notes, ainsi qu'une bibliothèque comprenant plusieurs dizaines d'effets audio, incluant notamment des filtres, des délais, des réverbérations et des unités de distorsion.

4. PERFORMANCES MUSICALES

Si *sequencer.party* offre un environnement de séquenceur collaboratif puissant, il sert également de plateforme entièrement modulaire pour les performances musicales en temps réel sur le Web. Au-delà du séquençage statique, son infrastructure prend en charge la construction et la manipulation dynamiques de chaînes de plugins complexes impliquant des générateurs de notes, des instruments, des effets et des modulateurs de paramètres, le tout en collaboration multi-utilisateurs et en temps réel.

Figure 4 : câblage typique de *sequencer.party*, un *step sequencer* WAM (à gauche) modulant certains paramètres de filtre d'un synthétiseur WAM.

Figure 5 : un modulateur orbiteur (en bas à droite), modulant certains paramètres du synthétiseur .

Figure 6 : piste/écran type dans *sequencer.party*. La piste 1 peut avoir un arrière-plan vidéo animé généré par des WAMs.

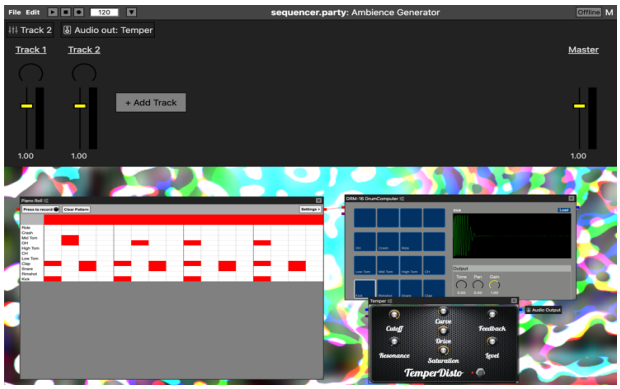

Figure 7 : en haut, le composant mixeur permet de régler le volume et le *panning* de chaque piste et de créer de nouvelles pistes.

4.1. Architecture orientée pistes et graphe de plugins

Dans *sequencer.party*, chaque session collaborative se déroule sur un « écran » virtuel qui ressemble conceptuellement à une piste dans un DAW traditionnel (Figure 6). Un écran occupe toute la surface de la fenêtre *sequencer.party* et les utilisateurs peuvent passer d'un écran virtuel à l'autre. Chaque écran contient un graphe de plugins (générateurs de notes, instruments, effets, modulateurs) connectés à une sortie audio. Un widget de mixage (également synchronisé entre participants) peut être utilisé pour régler le volume et le *panning* du son produit par chaque écran/piste (Figure 7). Chaque modification du tempo global, du mixage et des composants individuels est répercutée dans la session de tous les participants.

Sur chaque écran, les participants peuvent ajouter/supprimer/déplacer et interconnecter divers composants WAM : générateurs de notes, tels que des séquenceurs ou des générateurs algorithmiques, des instruments virtuels, notamment des synthétiseurs et des échantillonneurs, des effets audio, égaliseurs, délais etc., des modulateurs, capables de contrôler les paramètres d'autres plugins (Figure 7). Ces composants peuvent être déplacés et connectés/déconnectés entre eux. Ils sont organisés dans une architecture de graphe, où les participants construisent collaborativement des flux de signaux. Des connexions peuvent être établies entre les sorties et les entrées MIDI/audio, ou entre les sources de modulation/contrôle et les paramètres cibles, offrant ainsi la flexibilité nécessaire à la composition et à l'improvisation en direct.

4.2. Synchronisation en temps réel à l'aide de CRDT

Pour prendre en charge la collaboration distribuée, *sequencer.party* utilise une synchronisation basée sur la synchronisation d'états et soutenue par un algorithme de type CRDT, populaire dans les éditeurs de collaboratifs comme Google Docs (*Conflict-Free Replicated Data Type*). Cette approche garantit que tous les participants ont une vue cohérente de la session, même lorsque plusieurs utilisateurs modifient des éléments simultanément.

Tous les changements structurels et comportementaux de la session, y compris l'instanciation des plugins et le câblage des connexions dans le graphe, les changements de paramètres, la logique de génération des notes, le tempo et l'état du transport (play/stop/pause), l'attribution des sons dans les échantillonneurs, le mixeur, etc. sont répliqués en temps réel pour tous les utilisateurs. L'algorithme CRDT garantit une cohérence finale (*eventual consistency*) tout en permettant une réactivité locale, ce qui est particulièrement important pour maintenir une expérience utilisateur fluide pendant les performances en direct [19].

Dans *sequencer.party*, les états des plugins WAM, la structure des graphes (connexions, positions) et les paramètres globaux sont représentés sous forme de documents synchronisés à l'aide de la bibliothèque Yjs[7], une implémentation efficace de CRDT en JavaScript[8]. Cette approche permet de propager uniquement les différences d'état (deltas), réduisant la bande passante et permettant la convergence automatique des sessions sans mécanisme de verrouillage. La latence perçue dépend principalement du réseau, tandis que la cohérence structurelle est garantie indépendamment de l'ordre ou du délai de réception des mises à jour. Dans la pratique, cette approche permet une interaction perçue comme quasi instantanée dans des conditions réseau standards.

4.3. Synchronisation locale du rythme

Comme *sequencer.party* se concentre exclusivement sur la musique électronique, le système évite d'avoir à échanger des flux audio continus. Au lieu de cela, les informations globales relatives au tempo et au rythme sont partagées via la synchronisation d'état, tandis que l'alignement des rythmes est effectué localement sur chaque client. Par exemple, si un séquenceur pas à pas déclenche un échantillonneur, l'échantillon (qui est déjà synchronisé en tant que ressource) sera lu en rythme avec l'horloge locale, ce qui permettra une lecture parfaitement synchronisée sur tous les clients sans avoir recours à un réseau à faible latence. Cette conception garantit la préservation de la structure rythmique, même en présence de latence et de gigue (*jitter*) réseau.

4.4. Modulation de paramètres en contexte distribué

Un élément particulièrement sophistiqué de *sequencer.party* est sa gestion de la modulation de paramètres, déjà évoquée. Lorsqu'un modulateur WAM est connecté à un paramètre cible (par exemple, la coupure du filtre d'un synthétiseur), la plateforme garantit que le comportement de modulation reste à la fois expressif et efficace. Pour ce faire, elle synchronise l'état du modulateur (par exemple, la forme d'onde, la profondeur, le taux) et *exclut le paramètre modulé de l'état synchronisé* du plugin cible. Ainsi, *la modulation se produit localement* sur chaque client, ce qui permet de conserver un contrôle haute résolution (à la fréquence d'échantillonnage) sans avoir à synchroniser les paramètres qui changent rapidement sur le réseau. Cette séparation intelligente des préoccupations permet aux schémas de modulation complexes (par exemple, le vibrato piloté par LFO ou les balayages de filtre) de rester précis et expressifs dans les sessions collaboratives.

5. RÔLES ET MODÈLES D'INTERACTION

Au-delà de ses capacités en tant qu'outil de composition collaborative, *sequencer.party* est principalement utilisé comme plateforme de

[7] https://github.com/yjs/yjs

[8] https://blog.kevinjahns.de/are-crdts-suitable-for-shared-editing/ donne plus d'informations sur les caractéristiques de performance dans Yjs.

performances musicales en direct. Ces performances sont généralement initiées par un *organisateur* principal, qui rassemble un groupe de collaborateurs afin de contribuer en temps réel à une session musicale partagée, basée sur un navigateur. Cette section explore la manière dont les rôles sont gérés, comment fonctionne l'interaction en temps réel et comment ces performances sont souvent diffusées en streaming et conservées pour une utilisation future.

5.1. Lancement de session et contrôle d'accès basé sur les rôles

Les sessions en direct commencent généralement lorsque l'organisateur crée une nouvelle session collaborative, générant un lien d'invitation unique. Ce lien est partagé avec les collaborateurs, qui peuvent rejoindre la session depuis n'importe où en utilisant uniquement un navigateur web. Lorsqu'une invitation est créée, l'organisateur peut attribuer au participant invité un rôle qui déterminera son niveau d'interaction : un i*nvité* peut observer la session en temps réel, y compris la sortie audio et les changements d'interface utilisateur, mais ne peut interagir avec aucun élément ; un *interprète* peut manipuler les paramètres des composants WAM, par exemple, les enveloppes de synthétiseur, les coupures de filtre, mais pas le routage ou la structure du signal ; enfin, un *éditeur* a un contrôle complet également appelé « contrôle de la structure » ») et peut ajouter, supprimer ou connecter des composants, modifier des plugins et éditer la logique générative ou le contenu des séquences. Il peut également créer/éditer/supprimer des sessions, des écrans/pistes et des ressources. Ce modèle de contrôle hiérarchique garantit la stabilité et la cohérence musicale de la session, en particulier lors des diffusions en direct.

5.2. Performances virtuelles et usages pédagogiques

Dans sa forme la plus courante, *sequencer.party* est utilisé pour des performances entièrement virtuelles reposant uniquement sur des plugins WAM et des technologies web. Aucun matériel externe n'est nécessaire, car toute la génération, le traitement et la synchronisation audio s'effectuent dans le navigateur à l'aide de l'API Web Audio.

La machine de chaque participant gère son propre rendu audio en temps réel, tandis que l'état de session partagé garantit la cohérence structurelle et paramétrique. Nous avons utilisé cette configuration avec des étudiants en master pour illustrer les concepts d'improvisation collective et de musique évolutive. Bien que l'utilisation de cette plateforme à des fins éducatives n'était pas l'objectif principal, le fait qu'elle soit accessible, ne nécessite aucune installation ni configuration spécialisée et soit hautement modulaire, avec des composants WAM plug-and-play, en a fait un choix idéal pour les performances en direct avec des étudiants à distance et des étudiants connectés depuis la salle de classe.

Elle s'est également avérée utile pour introduire des concepts de synthèse musicale, tels que les composants modulaires d'un synthétiseur analogique, et pour démontrer le traitement des effets audio ou la modulation des paramètres. La fonctionnalité collaborative facilite l'interaction en classe, l'ordinateur de l'organisateur étant connecté à un vidéoprojecteur pour une visualisation partagée, tandis que les étudiants (locaux et à distance), équipés d'écouteurs, manipulent les synthétiseurs et les effets en temps réel. Seul le son provenant de l'ordinateur de l'instructeur est diffusé par les haut-parleurs de la salle de classe, ce qui garantit la clarté tout en permettant une exploration individualisée.

5.3. Édition musicale collaborative en temps réel

Pendant une performance, les contributeurs peuvent manipuler les paramètres du synthétiseur, ajuster la tonalité ou la modulation en temps réel, éditer des séquences de pas ou ajouter/supprimer des patterns de batterie à l'aide du séquenceur pas à pas ou de l'éditeur de type *piano roll*, utiliser différents comportements génératifs dans le *Function Sequencer*, et même participer à de simples sessions de *live coding* orchestrées par l'enseignant/organisateur, grâce à l'éditeur de code source Monaco intégré au séquenceur de fonctions, façonner des chaînes d'effets ou assigner des modulateurs aux paramètres des instruments. Toutes les modifications sont propagées à l'ensemble des participants, ce qui permet une édition fluide et simultanée sans conflits ni pics de latence. L'algorithme de synchronisation CRDT utilisé est également capable de gérer les pertes temporaires de connectivité, car il conserve l'historique des modifications, de la même manière que Google Docs gère l'édition de documents dans de tels cas.

5.4. Streaming, archivage et continuité des sessions

Outre les usages pédagogiques, la nature performative de *sequencer.party* permet la diffusion en direct des sessions via des plateformes telles que Twitch, YouTube ou des configurations basées sur OBS. Comme les performances se déroulent entièrement dans le navigateur, il est possible de toucher facilement un large public.

La plateforme prend également en charge l'enregistrement et la restauration des sessions, ainsi que la gestion des préréglages et des ressources, permettant de conserver et de réutiliser des configurations de performance complexes. Cette persistance facilite l'évolution des projets dans le temps, aussi bien pour les performances que pour les processus de composition et de répétition.

6. PERFORMANCES HYBRIDES

Si *sequencer.party* est une plateforme de collaboration musicale basée sur un navigateur, il permet également des configurations de performances

hybrides plus complexes *qui intègrent des instruments physiques et des environnements réels.* Dans cette configuration, les éléments de concert live, tels que les synthétiseurs matériels, les pédales d'effet, les consoles de mixage et la présence du public sont combinés avec les capacités modulaires à distance de *sequencer.party*, créant ainsi un contexte de performance *multimodal*.

6.1. Performance hybride, infrastructure de diffusion

Dans une performance hybride classique, l'artiste principal est physiquement présent dans une salle de concert ou un studio, équipé d'instruments matériels, de contrôleurs MIDI et d'unités d'effets. Un ordinateur exécutant *sequencer.party* sert de centre de coordination et peut également servir d'interface avec du matériel externe. Un grand écran derrière l'artiste affiche l'interface utilisateur de la session en direct, offrant transparence et contexte au public présent sur place et en ligne. Un ordinateur supplémentaire équipé du logiciel OBS (Open Broadcaster Software) agrège le son provenant de la table de mixage, les images provenant des caméras ou des captures d'écran et diffuse l'intégralité du spectacle sur des plateformes telles que YouTube Live ou Twitch. Cette configuration fait le pont entre les environnements de performance live traditionnels et le monde participatif et distribué de la création musicale sur navigateur.

6.2. Pont MIDI et intégration matérielle en temps réel

L'un des éléments clés de la fonctionnalité hybride de *sequencer.party* est la disponibilité de WAM prenant en charge les E/S MIDI. Ces modules permettent au navigateur *d'envoyer des événements MIDI* aux instruments matériels connectés (par exemple, déclencher des notes sur un synthétiseur externe), *de recevoir des messages de contrôle MIDI* provenant de contrôleurs ou de pédales matériels, et d'agir comme des couches de traduction entre le contenu généré par le logiciel et les appareils physiques produisant du son. De plus, *des contrôleurs WAM dédiés* ont été développés pour des synthétiseurs spécifiques, notamment le Korg MicroKorg, le Roland JX-3P et un Oberheim OB-6.

Ces WAM disposent d'*interfaces utilisateur graphiques inspirées du matériel d'origine*, avec des boutons, des curseurs et des commutateurs. Lorsqu'ils sont utilisés dans une session, même les collaborateurs à distance peuvent interagir avec ces interfaces utilisateur pour *modifier en temps réel les paramètres sonores des machines physiques*, accédant ainsi à la configuration locale de l'interprète depuis n'importe où dans le monde.

6.3. Participation sur site et à distance

L'un des aspects les plus intéressants des performances hybrides est la diversité des rôles et des emplacements des participants : l'interprète principal contrôle à la fois le matériel et les logiciels, agissant comme le nœud central de la performance. Les participants à distance (musiciens, codeurs, artistes visuels) peuvent se joindre à la session *sequencer.party* et modifier de manière collaborative les paramètres de synthétiseurs, ajuster les séquences ou le comportement du code, ajouter ou rediriger des effets. Les participants sur place peuvent également participer directement en se connectant à la session à l'aide d'ordinateurs portables, de téléphones ou de tablettes, contribuant ainsi de manière musicale ou interactive pendant la performance. Ce modèle de participation à plusieurs niveaux introduit une riche interaction entre la présence physique, la collaboration numérique et la création en temps réel.

7. ÉTUDE DE CAS : PERFORMANCE COLLABORATIVE EN CONTEXTE PÉDAGOGIQUE

Afin d'évaluer *sequencer.party* dans un contexte pédagogique réel et d'observer les modalités de son appropriation par des utilisateurs, une expérimentation a été menée en 2026 dans le cadre d'un cours consacré aux technologies Web Audio auprès de douze étudiants du Master Erasmus Mundus DIGICREA. L'objectif consistait à concevoir et interpréter collectivement une performance musicale collaborative d'une durée d'environ vingt minutes reposant exclusivement sur la plateforme. La vidéo de la performance ainsi que l'interview à chaud des élèves est disponible en ligne[9].

Les étudiants ont été répartis en quatre groupes de trois personnes, chacun responsable d'un écran correspondant à une piste fonctionnelle dans l'environnement (Figure 9). Chaque groupe s'est vu attribuer un rôle musical spécifique :

- piste 1 : responsable de l'animation graphique, source webcam ou MilkDrop avec une synchronisation sur une source audio, affichage en fond d'écran de la track 1 ;
- piste 2 : responsable d'une boucle de batterie produite par un step-sequenceur et un sequenceur piano-roll connectés à un sampleur et à divers effets ;
- piste 3 : (Figure 8) responsable de l'atmosphère générale, production de nappes et drones avec cinq instances d'un synthétiseur type Prophet V, un module de sons General MIDI, un générateur de bruit blanc, le tout piloté par un séquenceur, des générateurs pseudo aléatoires avec une chaine d'effets en sortie (distorsion, réverbération)
- piste 4 : produit la ligne mélodique sur deux synthétiseurs, grâce à un séquenceur et un module pseudo aléatoire.

Durant la semaine précédant la performance, les étudiants avaient progressivement construit leur installation musicale, expérimenté différentes configurations de graphes de plugins, répété collectivement et structuré leur composition (Figure 10).

Du point de vue de l'interaction homme-machine, ce processus peut être interprété comme la construction progressive d'un artefact numérique partagé servant simultanément d'instrument, de partition et d'espace de performance.

[9] https://youtu.be/BH1rWFaq_S8

Figure 8 : une vue simplifiée du patch instrumental pour la piste 3.

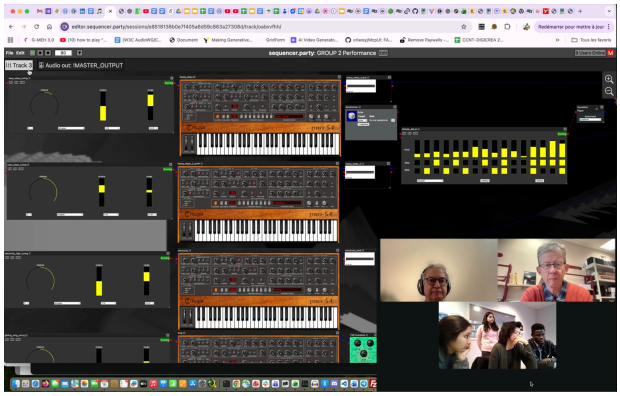

Figure 9 : performance musicale collaborative réalisée par douze étudiants du Master DIGICREA.

FORM of the IMPRO:

1. 0:10 : Hit the play button
2. 0:10 - 0:40 : risers intro
3. 0:40 - 1:20 : drums get inside
4. 1:20 - 4:00 : build (step by step: bass, leads, chords, 2nd lead)
5. 4:00 - 6:00 : development (changes, variations)
6. 6:00 - 8:00 : bridge (less)
   - turn down the drums
   - just chords
   - then a lead melody
   - simple bass (one-two pitches)
7. 8:00 - 10:00 : full power again
8. 10:00 - 11:00 small bridge
9. 11:00 - 12:00 power again
10. 12:00 - 14:00 : slowly rise of white noise (on separate track) and other instruments are slowly distorted more and more
11. 14:00 : ending with just noise
12. 15:00 end **(master fader down: Andrzej!)**

Figure 10 : structure de la composition du groupe n°2.

Ce type d'environnement correspond à ce que Shneiderman désigne comme un *creativity support tool* [24], c'est-à-dire un système interactif visant à amplifier les capacités créatives de ses utilisateurs plutôt qu'à automatiser la tâche créative elle-même.

Appropriation, apprentissage et instrumentalisation: les observations, complétées par des entretiens semi-directifs et les rapports des étudiants, révèlent un processus d'appropriation en plusieurs phases. Une phase initiale d'exploration, caractérisée par une charge cognitive relativement élevée liée à la compréhension du modèle modulaire et collaboratif, a été suivie d'une phase d'instrumentalisation, durant laquelle les étudiants ont commencé à utiliser le système comme un instrument expressif plutôt que comme un simple outil technique. Ce phénomène correspond au processus d'instrumentalisation décrit dans les théories de l'activité, dans lequel un artefact devient un instrument à travers l'usage [21]. Plusieurs étudiants ont explicitement mentionné une augmentation progressive de leur sentiment de maîtrise et de contrôle créatif. Un résultat particulièrement notable concerne l'utilisation spontanée des capacités programmables de certains plugins. Bien que l'API correspondante n'ait pas été enseignée[10], plusieurs étudiants ont utilisé l'assistant d'intelligence artificielle Gemini 3 (Google Deepmind) pour générer des scripts originaux à partir d'exemples existants et de descriptions textuelles. Cette pratique peut être interprétée comme une forme émergente de co-création humain-IA, dans laquelle l'utilisateur conserve le contrôle intentionnel tout en déléguant une partie de l'implémentation technique [18].

Du point de vue pédagogique, ce résultat est particulièrement significatif, car il suggère que la barrière technique à l'entrée peut être réduite sans limiter la complexité des artefacts produits, favorisant ainsi une exploration créative plus large.

Créativité distribuée, engagement et état de flow : la performance a également mis en évidence l'émergence de formes de créativité distribuée. Le système de synchronisation en temps réel a permis aux participants d'observer immédiatement les actions des autres et d'adapter leur propre contribution en conséquence, favorisant un processus de co-création dynamique. Ce mode d'interaction correspond au modèle de cognition distribuée, dans lequel le processus cognitif est réparti entre plusieurs individus et le système technique lui-même [16]. Plusieurs étudiants ont rapporté un fort sentiment d'immersion et d'engagement, certains décrivant l'outil comme « une drogue dure ». Cette observation qualitative peut être rapprochée du concept *d'état de flow* décrit par Csikszentmihalyi [9], caractérisé par une immersion profonde, une focalisation de l'attention et un sentiment de contrôle. L'équilibre entre complexité du système et capacités des utilisateurs semble avoir favorisé l'émergence de cet état. Ce résultat est particulièrement important dans un contexte pédagogique, car les états de flow sont associés à une amélioration de l'apprentissage et de la motivation intrinsèque.

Continuité pédagogique et agentivité technique : la possibilité d'intégrer des plugins WAM développés par les étudiants eux-mêmes dans des cours antérieurs, notamment à l'aide du langage FAUST, a contribué à renforcer leur sentiment d'agentivité technique. Les étudiants n'étaient plus seulement des utilisateurs du système, mais également des contributeurs à son écosystème. Cette continuité entre apprentissage technique et pratique artistique correspond aux approches pédagogiques constructivistes, dans lesquelles les apprenants construisent activement leurs propres outils et connaissances.

Limitations observées et implications pour la conception : la limite actuelle du nombre d'utilisateurs simultanés a occasionnellement perturbé l'accès à la session, soulignant l'importance de la scalabilité dans les environnements collaboratifs synchrones. Certains scripts générés par IA ont provoqué des instabilités, révélant un compromis entre extensibilité et robustesse. La stratégie adoptée par les étudiants, consistant à séparer les sessions d'expérimentation et de production,

[10] API du Function Sequencer: https://tinyurl.com/mr2d79cv

constitue une forme d'appropriation organisationnelle du système. Le mixeur intégré, limité à un contrôle par piste, a conduit les étudiants à implémenter leurs propres mécanismes de contrôle de gain au sein des graphes. Ce comportement illustre une appropriation créative, dans laquelle les utilisateurs adaptent le système à leurs besoins.

Ces observations, bien que qualitatives, confirment que l'approche fondée sur la synchronisation d'état et les architectures modulaires permet une appropriation rapide et une utilisation expressive dans un contexte réel, suggérant la pertinence de ce paradigme pour la conception d'environnements de création musicale collaborative.

8. CONCLUSION

Sequencer.party démontre le potentiel unique du navigateur en tant que plateforme pour la création et l'interprétation musicales collaboratives en temps réel, en particulier dans le domaine de la musique électronique.

Au cœur de son architecture se trouve la norme Web Audio Modules (WAM), qui fournit une base modulaire, extensible et performante pour la création de systèmes audio complexes dans l'environnement web. L'intégration des WAM et de leurs extensions, notamment celles qui prennent en charge l'édition collaborative, la gestion des ressources et le contrôle du matériel, s'est avérée être un choix pertinent pour la conception d'un environnement de création musicale distribué.

Au-delà de la description technique, cet article apporte trois contributions principales : (1) la présentation de *sequencer.party* comme une plateforme collaborative musicale basée sur les Web Audio Modules et un mécanisme de synchronisation d'état de type CRDT, (2) la démonstration de sa capacité à supporter des performances musicales collaboratives réelles, y compris dans des contextes hybrides combinant participants locaux et distants, et (3) une étude empirique de son appropriation dans un contexte pédagogique, mettant en évidence son potentiel comme outil de soutien à la créativité et à l'apprentissage.

L'expérimentation menée avec les étudiants du Master DIGICREA apporte des éléments empiriques complémentaires confirmant la pertinence de cette approche dans un contexte pédagogique et créatif. Elle met en évidence une appropriation rapide du système par des utilisateurs non spécialistes de la programmation, un fort soutien aux processus créatifs individuels et collectifs, ainsi qu'un engagement élevé des participants, caractéristique des états de flow. Elle révèle également l'émergence de pratiques hybrides combinant création artistique et génération de code assistée par intelligence artificielle. Ces résultats mettent en évidence le potentiel de *sequencer.party* comme environnement de soutien à la créativité et à l'apprentissage expérientiel. *Ils restent cependant exploratoires et nécessitent une validation quantitative à plus grande échelle.*

Associé à un mécanisme de synchronisation d'état fondé sur des CRDT, *sequencer.party* permet une collaboration musicale distribuée en temps réel, persistante et sans conflit, tout en garantissant la cohérence finale des sessions partagées. En combinant traitement audio avancé, modularité, programmabilité et collaboration en réseau, la plateforme démontre la viabilité d'environnements de création musicale collaboratifs puissants et accessibles directement dans le navigateur, et ouvre de nouvelles perspectives pour la conception d'instruments musicaux distribués, programmables et nativement collaboratifs sur le Web.

9. TRAVAUX FUTURS

L'expérimentation présentée dans cet article, menée avec des étudiants du Master Erasmus Mundus DIGICREA, fournit des observations qualitatives encourageantes concernant l'utilisabilité, l'appropriation et le potentiel créatif de *sequencer.party*. Cependant, elles nécessitent d'être complétées par des évaluations quantitatives formelles. Des travaux futurs viseront ainsi à évaluer plus rigoureusement l'utilisabilité de la plateforme à l'aide d'instruments standardisés tels que le *System Usability Scale* (SUS) [17], qui permet de mesurer la facilité d'utilisation perçue, ainsi que le *Creativity Support Index* (CSI) [8], afin d'évaluer dans quelle mesure la plateforme soutient les processus créatifs. Des outils d'évaluation de la collaboration, tels que le *Networked Minds Social Presence Inventory* [1], pourront également être utilisés afin de mesurer la perception de la présence sociale et la qualité de l'interaction entre participants. Enfin, la charge cognitive associée à l'utilisation du système pourra être analysée à l'aide du *NASA Task Load Index* (NASA-TLX) [15].

Au-delà de ces aspects liés à l'utilisabilité et à l'expérience utilisateur, des travaux futurs porteront également sur l'analyse des nouvelles formes de pratiques musicales rendues possibles par ce type d'environnement, notamment celles impliquant la programmation collaborative et l'intégration d'assistants d'intelligence artificielle dans les processus de création musicale. L'observation menée dans le cadre du Master Erasmus Mundus DIGICREA suggère que ces outils favorisent l'émergence de nouvelles formes d'interaction entre musiciens et systèmes logiciels, dans lesquelles la frontière entre programmation et pratique musicale devient plus perméable.

Sur le plan technique, des améliorations sont également envisagées afin d'augmenter la robustesse et la flexibilité de la plateforme, notamment en ce qui concerne la gestion des erreurs dans les plugins programmables et l'amélioration des mécanismes de gestion des sessions collaboratives.

Enfin, des travaux futurs porteront sur l'amélioration de la scalabilité du système, en particulier la gestion d'un plus grand nombre d'utilisateurs simultanés, afin de permettre des performances collaboratives

impliquant des ensembles de plus grande taille. Des améliorations de l'interface utilisateur sont également envisagées, notamment en ce qui concerne la gestion des rôles et les retours visuels sur les actions des participants, afin de faciliter l'utilisation du système dans des contextes pédagogiques et performatifs plus larges.

Ces perspectives visent à renforcer *sequencer.party* en tant que plateforme de référence pour la création musicale collaborative sur le Web, et comme environnement expérimental pour l'étude des nouvelles formes de création musicale distribuée.

10. REMERCIEMENTS

Ce travail a été soutenu par le gouvernement français, dans le cadre du plan d'investissement France 2030 géré par l'Agence nationale de la recherche, au titre du projet « UCA DS4H », référence ANR-17-EURE-0004.

11. RÉFÉRENCES

[1] Biocca and C. Harms, "Networked minds social presence inventory:(scales only, version 1.2). Measures of co-presence, social presence, subjective symmetry, and intersubjective symmetry," 2003.

[2] M. Buffa, J. Lebrun, S. Ren, S. Letz, Y. Orlarey, R. Michon, and D. Fober. Emerging W3C APIs opened up commercial opportunities for computer music applications. In The Web Conference 2020 DevTrack. Taipei, Taiwan. 2020.

[3] M. Buffa, S. Ren, O. Campbell, J. Kleimola, O. Larkin, and T. Burns. 2022. Web Audio Modules 2.0 : An Open Web Audio Plugin Standard. In WWW '22 : The ACM Web Conference. 2022. ACM, France, pp 364–369. 2022.

[4] M. Buffa and A. Vidal-Mazuy. "WAM-studio, a Digital Audio Workstation (DAW) for the Web." Companion Proceedings of the ACM Web Conference 2023.

[5] M. Buffa, D. Girard, & A. Hofr, Using Web Audio Modules for Immersive Audio Collaboration in the Musical Metaverse. Proceedings of the IEEE 5th International Symposium on the Internet of Sounds (pp. 1-10). Erlangen, Germany. 2024.

[6] M. Buffa, S. Ren, T. Burns, A. Vidal-Mazuy & S. Letz. Evolution of the Web Audio Modules Ecosystem. Proceedings of the Web Audio Conference, Lafayette, USA. 2024.

[7] M. Buffa et al. Ten years of Web Audio Module: audio plugins for the web. Journal of the Audio Engineering Society (JAES). Vol. 73, no. 12, pp. 804-826, Dec. 2025.

[8] E. Cherry and C. Latulipe, "Quantifying the creativity support of digital tools through the creativity support index," ACM Transactions on Computer-Human Interaction (TOCHI), vol.21, no.4, pp.1–25, 2014.

[9] M. Csikszentmihalyi. Flow: The Psychology of Optimal Experience. 1990.

[10] R. B. Dannenberg, H. Zhang, A.K Meena, A. Joshi, A. Kuran, & J. Sastre. Collaborative Music Creation and Performance with Soundcool Online. Proceedings of the Web Audio Conference (WAC), Barcelona, Spain (pp. 5-7). 2021.

[11] D. Dziwis. PdXR-the Evolution of Pure Data into the Metaverse. Proceedings of the International Computer Music Conference ((pp. 1-8). 2023.

[12] D. Dziwis, H. Von Coler and C. Pörschmann, Orchestra: a toolbox for live music performances in a web-based metaverse. Journal of the Audio Engineering Society, 71(11), pp.802-812. 2023.

[13] D. Dziwis, H. Von Coler and C. Pörschmann, October. Live Coding in the Metaverse. In 2023 4th International Symposium on the Internet of Sounds (pp. 1-8). 2023.

[14] X. Favory, X. Serra, Multi Web Audio Sequencer : Collaborative Music Making. Proceedings of the Web Audio Conference, Berlin, Germany. 2018.

[15] S. G. Hart, "NASA-Task Load Index (NASA-TLX); 20 years later." Proceedings of the human factors and ergonomics society annual meeting. Vol. 50. No. 9. Sage CA: Los Angeles, CA: Sage publications, 2006.

[16] E. Hutchins, Cognition in the Wild. MIT press. 1995.

[17] J. R. Lewis, "The system usability scale: past, present, and future," International Journal of Human–Computer Interaction, vol. 34, no. 7, pp. 577–590, 2018.

[18] T. Lubart. How can computers be partners in the creative process: classification and commentary on the special issue. International Journal of Human-Computer Studies, 63(4-5), pp.365-369. 2005.

[19] P. Nicolaescu, K. Jahns, M. Derntl, & R. Klamma. Near real-time peer-to-peer shared editing on extensible data types. Proceedings of the ACM International Conference on Supporting Group Work (pp. 39-49). 2016.

[20] E. Paris. "Une approche du patching audio collaboratif." Thèse de doctorat en musicologie, Univ. Paris 8, 2018.

[21] Rabardel, P., Les hommes et les technologies; approche cognitive des instruments contemporains (p. 239). Armand Colin. 1995.

[22] S. Ren, L. Pottier, and M. Buffa. "Build webaudio and javascript web applications using jspatcher: A web-based visual programming editor." In Web Audio Conference 2021.

[23] C. Rottondi, C. Chafe, C. Allocchio, & A. Sarti, An overview on networked music performance technologies. IEEE Access, 4, 8823-8843. 2016.

[24] B. Shneiderman, 2007. Creativity support tools: accelerating discovery and innovation. Communications of the ACM, 50(12), pp.20-32.

[25] L. Turchet, and S.Krstulović. "DSP as a Service: Foundations and Directions." IEEE Open Journal of the Communications Society, 2024.