



HAL
open science

Les instruments de musique numériques pour les personnes en situation de handicap cognitif et mental : exemples d'applications via le web

Matteo Olivo

► To cite this version:

Matteo Olivo. Les instruments de musique numériques pour les personnes en situation de handicap cognitif et mental : exemples d'applications via le web. Sound and Music Computing, Laboratoire ECLLA - Université Jean Monnet Saint-Etienne; INRIA; GRAME CNCM Lyon, Jun 2022, Saint-Etienne (FR), France. <10.5281/zenodo.6822203>. <hal-04941042>

HAL Id: hal-04941042

<https://hal.science/hal-04941042v1>

Submitted on 11 Feb 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

LES INSTRUMENTS DE MUSIQUE NUMERIQUES POUR LES PERSONNES EN SITUATION DE HANDICAP COGNITIF ET MENTAL : EXEMPLES D'APPLICATIONS VIA LE WEB

Matteo Olivo

Laboratoire ECLLA - Université
de Saint-Etienne/Université de
Lyon
matteo.olivo@univ-st-etienne.fr

RÉSUMÉ

L'article aborde le sujet des instruments de musique numériques pour les personnes en situation de handicap cognitif et mental. Dans la première partie je passerai en revue les principaux types d'instruments de musique numériques adressés à toutes les formes de handicap, et les définitions utilisées dans la littérature pour les identifier. Cette analyse du contexte me permettra de mieux situer et décrire, ensuite, les applications que j'ai développées dans le cadre de mon doctorat au sein du département de musicologie de l'Université Jean Monnet de Saint-Etienne, codirigé au laboratoire Parcours Santé Systémique de l'Université Lyon 1. La recherche concerne la conception de programmes pédagogiques et d'instruments de musique numériques pour le développement des compétences psychosociales et musicales des personnes en situation de handicap cognitif et mental¹. En particulier, je décrirai les adaptations apportées pour rendre les outils accessibles, le processus de design suivi et l'implémentation informatique de deux applications. Enfin, j'analyserai les résultats des expérimentations menées et je définirai les perspectives futures de la recherche.

1. LES INSTRUMENTS DE MUSIQUE NUMERIQUES INCLUSIFS

1.1 Etat de la recherche et terminologie utilisée

Depuis des décennies, les technologies musicales et audiovisuelles sont de plus en plus diffusées dans le domaine du handicap de toutes formes (moteur, mental, cognitif, sensoriel et psychique). En effet, ces dispositifs numériques permettent de créer des instruments de musique adaptés aux besoins et capacités de l'utilisateur, en démocratisant ainsi l'accès à la pratique musicale et en

ouvrant de nouvelles possibilités expressives. En raison de leur accessibilité et adaptabilité, ces outils trouvent une large variété d'applications : dans des contextes pédagogiques formels et informels [16], dans le cadre de performances musicales de niveau amateur et avancé² [9], et aussi au sein de parcours thérapeutiques, comme par exemple dans les programmes de musicothérapie ou d'autres initiatives visant à améliorer les capacités cognitives, motrices ou le bien-être des personnes en situation de fragilité [2][8].

Si dans la littérature concernant l'informatique musicale le thème des instruments de musique pour les personnes en situation de handicap n'a pas encore été beaucoup exploré [6], on peut toutefois constater un intérêt croissant pour le développement et l'analyse de ce type de dispositifs de la part des institutions de recherche, ainsi que des entreprises et des organisations caritatives³. En effet, grâce au protocole MIDI, qui a permis une transmission rapide et simple des commandes musicales, à partir des années 1980 beaucoup d'instruments de musique « adaptés » ont été conçus, comme par exemple le *Soundbeam*⁴, dispositif basé sur un faisceau sonore⁵ qui déclenche des événements MIDI lorsqu'il est coupé, ou le *Magic Flute*⁶, instrument qui permet de jouer des notes MIDI à l'aide d'un capteur de pression respiratoire. Ils représentent des exemples des nombreuses possibilités créatives que ce protocole de communication a ouvert dans ce domaine [6]. L'attention croissante pour ce type d'instruments est mise en évidence par le fait que le sujet est de plus en plus abordé dans le cadre d'importantes conférences du domaine de l'informatique musicale, comme par exemple la *NIME (New Interfaces for Musical Expression)* et la *SMC (Sound and Music Computing)*. De plus, des conférences spécifiques sur ce thème sont apparues, comme la *OHMI - Music Making for People with Disability*, entièrement dédiée aux instruments de musique pour les personnes ayant un

¹ Les différentes catégories de handicap ont été définies par la loi n. 2005-102 du 11/02/2005, intitulée « *Loi pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées* ».

² Mc Pherson, Morreale et Harrison inscrivent les instruments utilisés dans ce contexte dans la catégorie des « *performance-focused instruments* ».

³ Voir, à ce propos, les organisations *Drake Music* (www.drakemusic.org, site consulté le 27/11/2021), *Open Up Music* (openupmusic.org, site consulté le 27/11/2021) ou *EyeHarp association* (eyeharp.org, site consulté le 3/12/2021).

⁴ www.soundbeam.co.uk [consulté le 2/4/2022].

⁵ Il s'agit d'un sonar, technologie née pour permettre la détection du mouvement à grande distance sous l'eau.

⁶ www.mybreathmymusic.com/en/glenn-2 [consulté le 2/4/2022].

handicap moteur. Cet événement prévoit aussi un prix pour les développeurs de dispositifs de ce type.

Selon Emma Frid [4], qui a effectué un travail de recherche à ce sujet, basé sur l'analyse de 113 documents publiés entre 1990 et 2018⁷, il n'y a pas encore une terminologie unique, communément reconnue par la communauté scientifique, concernant les instruments de musique inclusifs. Dans la publication issue de cette recherche, Frid utilise l'expression *Accessible Digital Musical Instruments* (ADMIs), dont elle fournit la définition suivante : « *Accessibles musical control interfaces used in electronic music, inclusive music practice and music therapy settings* » [5]. L'expression *Accessible Instruments* est aussi utilisée par d'autres auteurs, parmi lesquels Harrison et McPherson [7]. Le terme *adaptive*, qui met en valeur la capacité de ces dispositifs d'être personnalisés en fonction des besoins de l'utilisateur, apparaît aussi souvent dans la littérature du domaine, avec des tournures différentes : par exemple, Graham-Knight et Tzanetakis [6] utilisent l'expression *Adaptive Music Technology*, en définissant cette notion de la manière suivante : « *The use of digital technologies to allow a person who cannot otherwise play a traditional musical instrument to play music unaided* ». Par contre, Vamvakousis et Ramirez [19] utilisent plutôt la formulation *Adaptive Digital Musical Instruments*. Enfin, dans certaines publications scientifiques du domaine, on trouve d'autres expressions comme *Inclusive music* et *Assistive music technology* [13].

1.2 Catégories de dispositifs existantes, technologies employées

Plusieurs auteurs ont essayé d'élaborer une classification des instruments de musique adressés aux personnes en situation de handicap. Parmi les travaux que j'ai pu examiner à ce sujet, les recherches de Frid mentionnées ci-dessus [4] me semblent être l'une des analyses les plus vastes du point de vue de la taille et de la variété de l'échantillon examiné, et ainsi fournir une image assez représentative de l'état de l'art actuel dans ce domaine : en effet, 83 instruments différents sont étudiés dans ce travail, concernant plusieurs formes de handicap (physique, cognitif, mental et sensoriel). La plupart des autres recherches publiées, par contre, ne concerne que les instruments pour des personnes en situation de handicap moteur (voir, par exemple, [6]). La classification a été effectuée sur la base du type d'interface et du mode principal d'interaction musicale de l'instrument. Les catégories recensées sont les suivantes :

- contrôleurs tangibles : objets physiques qui peuvent être touchés ou mis en mouvement, pour produire du son (voir, par exemple, l'instrument *WamBam*) ;
- contrôleurs sans contact : dispositifs qui utilisent des capteurs ou des caméras pour détecter les mouvements du corps dans l'espace ; les mouvements ne doivent pas

impliquer le fait de toucher activement des objets (comme par exemple dans le cas du *SoundBeam*) ;

- instruments adaptés : instruments existants qui ont été modifiés ou augmentés, comme par exemple les *hyperinstruments* (voir, à ce propos, la basse adaptée analysée par Harrison et McPherson [7]) ;

- interfaces musicales *Brain-Computer* : dispositifs qui utilisent des capteurs biométriques comme les électrodes pour monitorer l'électroencéphalographie, les EEG, etc. (voir, par exemple, le dispositif *Biomuse*) ;

- contrôleurs du regard : instruments qui utilisent le système de *eye-tracking* (comme le *EyeHarp*, analysé par Vamvakousis et Ramirez [19]) ;

- contrôleurs à écran tactile ;

- contrôleurs audio : instruments qui utilisent le microphone comme source sonore ;

- interfaces actionnées par la bouche : contrôleurs de vent qui utilisent des capteurs de respiration (comme le *Magic flute* mentionné ci-dessus) ;

- contrôleurs portables et prothèses : dispositifs qui peuvent être portés ou utilisés comme une extension du corps du musicien (voir, par exemple, le *HipDisk*) ;

- interfaces contrôlées par la souris de l'ordinateur (voir, à ce propos, le dispositif *MidGrid*).

Parmi les 83 instruments analysés, 39 produisent une réponse monomodale (principalement sonore), 40 bimodale (auditive-visuelle dans 33 cas et auditive-vibrotactile dans 7 cas) et 4 trimodale (auditive-visuelle-vibrotactile). Les types de capteurs les plus utilisés pour le développement de ces instruments sont les capteurs tactiles, les accéléromètres et les caméras. Les instruments recensés sont souvent réalisés en utilisant comme base des circuits créés avec des microcontrôleurs, ou d'autres technologies commerciales comme la Nintendo *Wii-mote*, la *Kinect* ou le capteur *Leap Motion*. Concernant les actionneurs, les plus utilisés sont les solénoïdes, les *faders* motorisés et les actionneurs de vibration. En général, les développeurs cherchent des technologies bon marché, mais accessibles et fiables.

2. LES APPLICATIONS MUSICALES CONÇUES DANS LE CADRE DE MON DOCTORAT DE RECHERCHE

2.1 Types d'applications développées et objectifs pédagogiques

L'objectif du projet de recherche est de concevoir des parcours pédagogiques, à l'aide d'outils numériques, visant au développement des compétences psychosociales (confiance en soi, communication verbale et non verbale, capacité de gérer les relations au sein d'un groupe, etc.) et musicales des personnes en situation de handicap cognitif et mental ; les déficits sensoriels ne sont pas analysés dans le cadre de la recherche et les handicaps moteurs ne sont visés que lorsqu'ils sont

⁷ Il s'agit d'actes des conférences NIME, SMC et ICMC - *International Computer Music Conference*, articles scientifiques, ouvrages et thèses de doctorat.

associés à des formes de déficience cognitive ou mentale (la dyspraxie, dont je parlerai dans la suite de l'article, est l'unique forme de handicap moteur prise en compte). Le projet s'adresse principalement aux jeunes (élèves du collège, adolescents en dehors du cadre scolaire), mais aussi aux adultes. Les compétences psychosociales sont des capacités dont l'acquisition contribue au bien-être et à la santé⁸ de la personne, et se fait principalement à travers des activités de groupe : la musique, en tant que pratique collective qui facilite la création de relations, peut représenter un véhicule assez efficace à ce propos. Les applications que je développe pour la réalisation de ces parcours doivent, donc, être accessibles et intuitives pour des personnes ayant des difficultés du point de vue cognitif et n'ayant pas d'expérience dans le domaine musical ; ces outils doivent permettre aux participants au projet de jouer immédiatement, sans passer par une acquisition graduelle de la maîtrise de l'instrument, pour créer une dynamique de groupe et garder la concentration [6]. Mc Pherson, Morreale et Harrison inscrivent ce type d'applications dans la catégorie des *therapeutic instruments*, qu'ils définissent de la manière suivante :

Accessible instruments [...] designed to elicit the therapeutic or wellbeing aspects of music making for disabled people with physical and cognitive impairments and learning difficulties, who may be identified as 'non-musicians'. Here we refer to the latter case as 'therapeutic instruments', referring to the design goals of enabling musicking for wellbeing purposes, and not necessarily solely for use in formal music therapy. [9].

Le choix de centrer la composition sur le thème du paysage sonore naturel est cohérent avec l'objectif décrit ci-dessus : en effet, travailler dans le domaine du son, et pas des notes, permet d'éviter de rentrer dans des aspects trop techniques et de passer immédiatement à une phase de création. Les applications développées simulent (en utilisant des échantillons enregistrés ou des techniques de synthèse sonore⁹) des bruits de la nature ou le son de certains instruments de musique (comme par exemple le djembé ou le didgeridoo) qui, dans l'imaginaire collectif, peuvent être associés à ce type d'environnement sonore. Même les simulations des instruments restent très intuitives et permettent de produire du son facilement.

En plus des compétences psychosociales, le projet vise (surtout pour les activités du projet qui se déroulent en milieu scolaire) à développer des compétences théoriques, en cohérence avec les programmes ministériels d'éducation musicale du cycle 3^{ème} et 4^{ème} : par exemple, l'apprentissage de certaines notions de base comme celles de fréquence, amplitude, rythme, la capacité à distinguer le timbre et les catégories

principales des instruments, et d'autres. Pour poursuivre ces objectifs je suis en train de développer une deuxième série d'outils numériques, avec la collaboration des professionnels de l'Éducation Nationale.

Olivier Nicolas a élaboré une classification des applications pédagogiques à utiliser dans le cadre de l'enseignement musical : 1) les applications en dehors du domaine musical, mais dont les fonctionnalités servent un but d'apprentissage (par exemple un outil pour faciliter la lecture des partitions aux personnes dyslexiques) ; 2) les applications musicales d'entraînement pur, catégorie subdivisée en deux sous-catégories : les exercices, plus centrés sur le contenu et ayant moins de distractions sonores et visuelles, et les *serious games*, plus ludiques et animés, et basés sur un système de niveaux qui rend la progression plus attractive ; 3) les applications pour l'interprétation, l'improvisation et la création [10]. En suivant cette classification, nous pouvons inscrire la première série d'applications numériques susmentionnée, visant à la réalisation de la composition autour du paysage sonore, dans la catégorie des outils pour l'interprétation, l'improvisation et la création ; les autres applications pédagogiques, pour l'apprentissage des notions musicales de base, appartiennent à la catégorie des outils pour l'entraînement pur, et notamment des exercices.

2.2 Technologies et environnements de programmation utilisés

Les applications¹⁰ sont développées principalement sous forme d'applications web en utilisant la Web Audio Api, une Api Javascript de haut niveau, pour le traitement de l'audio à basse latence et la synthèse sonore dans l'environnement web¹¹. Le processus de traitement du signal audio a lieu au sein d'un système modulaire (*AudioContext*) conçu par le *World Wide Web Consortium* (W3C) sous forme de graphe audio. Dans cet environnement, des nœuds audio (*AudioNodes*), ayant la fonction de producteurs, transformateurs ou analyseurs du signal, sont connectés entre eux et branchés ensuite à la sortie de l'application. La Web Audio Api fournit une série de nœuds audio natifs qui implémentent les fonctions les plus communes (lecteurs de *buffers*, filtres, spatialisation du son, convolution, etc.) [3]. Cependant, pour pouvoir créer des algorithmes de traitement du signal plus complexes, il est aussi possible de développer des nœuds personnalisés (*AudioWorkletNodes*)¹², codés en Javascript directement ou dans le langage *WebAssembly*. Ce dernier peut être généré par des *Domaines Specific Languages (DSL)*, comme par exemple Faust. Faust (*Functional Audio Stream*) est un langage fonctionnel, synchrone, spécifiquement conçu par le Grame (Centre National de Création Musicale à

⁸ Selon la définition fournie par l'Organisation Mondiale de Santé, la santé est conçue de manière holistique, c'est-à-dire comme un état de bien-être physique, mental et social, et pas uniquement comme l'absence de maladie.

⁹ Comme dans le cas de l'application « La mer et les vagues », décrite dans le paragraphe 3.1.

¹⁰ <https://www.matteoolivo.com/>.

¹¹ <https://www.w3.org/TR/webaudio/#introductory> [consulté le 23/12/2021].

¹² <https://developers.google.com/web/updates/2017/12/audio-worklet> [consulté le 23/12/2021].

Lyon) pour le traitement audionumérique en temps réel et la synthèse sonore [3]. Le compilateur de ce langage permet, en utilisant le *transpiler* Emscripten, basé sur la technologie *LLVM - Low Level Virtual Machine*, de traduire le code Faust directement dans le langage *WebAssembly*, à partir du fichier `dsp`¹³ [11]. Une fois le répertoire où se trouve le fichier `dsp` sélectionné sur le terminal, il suffit d'écrire le script suivant :

```
faust2wasm -worklet osc.dsp
```

Cette commande génère un fichier *WebAssembly* en format binaire (`osc.wasm`), et un fichier Javascript (*wrapper*) nommé `osc.js`, qui permet de transformer le code `dsp` en nœud audio, qui pourra être intégré dans le script Javascript de l'application et connecté à d'autres nœuds¹⁴. Le nom du fichier `dsp` est utilisé pour définir le constructeur du *AudioWorkletNode* (Faustosc). La fonction *startosc* décrite ci-dessous permet de créer le nœud, récupérer la description JSON de la UI et les parcours pour accéder aux paramètres `dsp`, et enfin de connecter le nœud à la sortie audio de la machine/dispositif. La méthode *setParamValue* assigne la valeur 0.6 au paramètre *freq*. La modification des paramètres peut être aussi contrôlée de manière dynamique dans le script Javascript, à travers un gestionnaire d'événements associé à un élément de l'interface web (un *slider*, un bouton, etc.).

```
function startosc(){
  var pluginURL = ".";
  var plugin = new
  Faustosc(audioContext, pluginURL);
  plugin.load().then(node => {
    osc_dsp = node;
    console.log(osc_dsp.getJSON());
    console.log(osc_dsp.getParams());
    osc_dsp.connect(audioContext.destination);
  });
}
node.setParamValue("/osc/freq",0.6);
```

Pour certaines des applications web que j'ai développées, j'ai utilisé uniquement les nœuds natifs de la Web Audio Api, tandis que pour d'autres j'ai créé des *AudioWorkletNodes* avec Faust, en suivant la procédure précédemment décrite. Cela m'a permis d'utiliser les bibliothèques (par exemple la bibliothèque sur la synthèse par modèles physiques) et les nombreuses fonctionnalités que cet environnement met à disposition, optimisées pour le traitement du signal audio. En plus de la Web Audio Api, j'ai aussi utilisé d'autres Api Javascript comme la Api *Sensor* (notamment le constructeur *Accelerometer*), pour récupérer les valeurs de l'accéléromètre¹⁵, et *MediaStream Recording*¹⁶ pour enregistrer le son capté

par le microphone des dispositifs. Enfin, en ce qui concerne le développement des interfaces, j'ai utilisé les langages traditionnels de la programmation web (HTML et CSS) et le *framework* Bootstrap qui permet de créer assez facilement des interfaces réactives, afin de pouvoir utiliser l'application sur différents types de supports.

Parallèlement aux applications web, j'ai aussi développé d'autres prototypes d'applications en Max/MSP¹⁷. En effet, par rapport à la Web Audio Api, qui a été créée plus récemment¹⁸, Max offre plusieurs avantages : d'un côté, ayant un plus long historique et une plus grande communauté de développeurs, il possède plus de fonctionnalités et permet de créer des outils plus complexes. De l'autre côté, Max/MSP est basé sur un système de programmation par objets graphiques qui, par rapport à la programmation textuelle de la Web Audio Api, est plus intuitif et permet de développer des prototypes plus rapidement. Une fois testés et validés, ces outils seront développés sous forme d'applications web.

2.3 Les adaptations apportées pour rendre les applications plus accessibles

Généralement, dans la littérature concernant les sciences de l'éducation, l'intégration des technologies numériques dans l'enseignement est citée comme un véhicule pour développer des compétences-clés, comme la pensée computationnelle ou la littératie en matière de médias [1]. Dans le cadre de ma recherche, le choix d'utiliser des applications à la place des instruments de musique traditionnels poursuit principalement un autre objectif, qui consiste à rendre ces outils pédagogiques plus accessibles et adaptables aux capacités cognitives et physiques des usagers. Le développement des compétences numériques représente un objectif secondaire du projet : en effet, les activités menées jusqu'à présent ont montré que la plupart des jeunes impliqués maîtrisent déjà assez bien les dispositifs numériques et sont capables d'en comprendre rapidement le fonctionnement, malgré leurs difficultés cognitives¹⁹.

Comme indiqué ci-dessus, le projet s'adresse aux personnes en situation de handicap mental ou cognitif, c'est-à-dire des individus ayant un développement intellectuel et une capacité d'apprentissage significativement inférieures à la moyenne, ou un dysfonctionnement des autres fonctions cognitives : l'attention, la mémoire, le langage, la capacité perceptive (gnosies), la capacité à exécuter des séquences coordonnées de gestes (praxies), la cognition sociale ; les troubles spécifiques de l'apprentissage (dyslexie, dyspraxie, dyscalculie, etc.) s'inscrivent également dans la catégorie du handicap cognitif. Compte tenu de

¹³ Emscripten génère le code *WebAssembly* à partir du code C++.

¹⁴ Voir, à ce propos, <https://github.com/grame-cncm/faust/tree/master-dev/architecture/webaudio> [consulté le 23/12/2021].

¹⁵ <https://developer.mozilla.org/en-US/docs/Web/API/Accelerometer> [consulté le 24/12/2021].

¹⁶ https://developer.mozilla.org/en-US/docs/Web/API/MediaStream_Recording_API [consulté le 24/12/2021].

¹⁷ https://www.matteoolivo.com/maxapplications_en.html

¹⁸ Le premier *working draft* de la Web Audio Api date du 25/11/2011 (<https://www.w3.org/TR/2011/WD-webaudio-20111121/>, page consultée le 24/12/2021), tandis que la première version de Max/MSP, qui intégrait la partie dédiée au traitement du signal (*Musical Signal Processing*), a été développée en 1997 par la société *Cycling74*.

¹⁹ Voir, à ce propos, le chapitre 4.

l'ampleur du public ciblé, il est très difficile de trouver des solutions transversales, qui peuvent faire face à tous les troubles mentionnés. Les adaptations apportées aux applications, donc, visent à compenser les déficits que l'on retrouve le plus souvent dans ce public, et qui peuvent impacter sur l'utilisation de ce type d'outils : principalement, les difficultés ciblées sont celles liées à la compréhension (dues au déficit mental), à la lecture, à la motricité fine et à la coordination des mouvements. Les aménagements apportés, décrits ci-dessous, ont été identifiés sur la base de l'analyse de la littérature du domaine, de certains documents publiés par le Ministère de l'Éducation Nationale sur le site Eduscol²⁰ et des ressources en ligne de la *WAI – Web Accessibility Initiative*²¹, mises à disposition par le W3C et concernant le développement et le design des sites web. Une confrontation avec plusieurs professionnels du domaine éducatif et médico-social a été également effectuée.

- Aménagements pour faciliter la compréhension du fonctionnement des applications : pour rendre les logiciels plus accessibles, la quantité de fonctionnalités et d'éléments graphiques sur l'interface a été réduite au minimum. De plus, les interfaces contiennent des informations écrites assez détaillées pour permettre aux utilisateurs, et aussi à leurs parents et aux professionnels qui les accompagnent, d'en comprendre facilement le fonctionnement. Enfin, le texte sur les interfaces est entièrement écrit en français, en évitant au maximum le vocabulaire technique, et sur les boutons des petites icônes ont été placées, qui permettent de mieux comprendre leur fonction (par ex. des icônes représentant le type de son déclenché). Le changement de couleur des boutons, lorsqu'ils sont activés, permet d'avoir un retour visuel de l'interaction effectuée. En ce qui concerne les applications web, l'accessibilité est encore plus marquée, étant donné que les applications ne nécessitent aucune installation de logiciels : il suffit de cliquer sur un site internet pour pouvoir y accéder.

- Aménagements pour faire face aux troubles liés à la lecture : tous les textes présents sur les interfaces ont été écrits avec la police *OpenDyslexic*, conçue pour faciliter la lecture aux personnes dyslexiques²². En plus, d'autres adaptations ont été apportées concernant la taille de la police, l'utilisation du caractère gras, l'interligne, le contraste des couleurs. Les explications concernant le fonctionnement des applications sont réalisées aussi en format audio, à travers une technique de synthèse vocale²³ basée sur l'utilisation de l'API Javascript *Web Speech*²⁴ ; la vitesse de lecture peut être réglée en fonction des besoins de l'utilisateur et la longueur du discours est réduite au minimum, de manière à fournir les informations essentielles et de permettre, en même temps, la compréhension du message. En effet, la courte

durée du texte permet d'éviter l'information transitoire, un phénomène qui a des effets négatifs sur l'apprentissage et qui se produit lorsqu'une modalité de présentation permanente (par exemple un texte écrit) est transformée en présentation transitoire (par exemple sous forme orale) [17]. Les mêmes explications sont aussi fournies sous forme textuelle, au sein d'un document qui s'affiche lorsqu'un bouton sur l'interface est pressé : ce texte est rédigé avec des couleurs différentes pour permettre de distinguer les syllabes, et une ligne sur deux est surlignée en couleur²⁵. Ces aménagements ont été apportés pour faciliter la lecture aux personnes dyslexiques et dyspraxiques ; la dyspraxie entraîne également des déficits visuo-spatiaux et peut provoquer des difficultés concernant la lecture²⁶.

- Aménagements pour faire face aux difficultés liées à la motricité fine et à la coordination des mouvements : la série d'instruments de musique numériques pour la composition autour du paysage sonore est composée d'applications pour le web et d'autres prototypes développés en Max/MSP. Ces derniers peuvent être contrôlés par des outils différents : des contrôleurs MIDI avec des *faders*, des *pads*, un clavier, la souris de l'ordinateur. De plus, en ce qui concerne les applications web, certains paramètres du son sont contrôlés par l'accéléromètre du dispositif : cela permet de les modifier en temps réel, de manière intuitive et à travers des gestes simples ; dans ces outils l'interface est réactive, de façon à pouvoir les utiliser sur plusieurs types de supports : ordinateur portable et fixe, tablette, *smartphone*. Cette variété d'outils permet à chaque utilisateur de choisir le type d'application, le support et le mode d'interaction les plus adaptés, en fonction des propres capacités motrices ; quel que soit le dispositif utilisé, le mode d'interaction reste toujours très simple et intuitif. L'adaptabilité à plusieurs types de supports permet aussi de réduire les inégalités concernant l'accès aux technologies numériques : en effet, il suffit d'avoir un *smartphone* et une connexion web, pour pouvoir jouer de la musique partout.

2.4 Le processus de design des applications et la méthode de recherche appliquée

Le projet a prévu la mise en place d'un système d'évaluation portant sur trois domaines : le développement des compétences psychosociales, l'acquisition des compétences musicales (objectif concernant principalement les activités en milieu scolaire), le niveau d'accessibilité des dispositifs numériques réalisés et leur impact sur la réalisation du projet. L'évaluation est basée sur la méthode qualitative et, notamment, sur l'observation participante des groupes

²⁰ <https://eduscol.education.fr/> [consulté le 26/12/2021].

²¹ <https://www.w3.org/WAI/> [consulté le 26/12/2021].

²² <https://opendyslexic.org/> [consulté le 24/12/2021].

²³ Au moment de la rédaction de l'article, seulement certaines applications ont été équipées du dispositif de la synthèse vocale, parmi lesquelles l'application « Djembé » : https://www.matteoolivo.com/app_djembeseq_fr.html.

²⁴ https://developer.mozilla.org/fr/docs/Web/API/Web_Speech_API [consulté le 28/03/2022].

²⁵ Les boutons sont visibles sur la figure 1 (paragraphe 3.2).

²⁶ Voir, à ce propos, le site du projet « Le cartable fantastique » : www.cartablefantastique.fr [consulté le 28/03/2022].

pendant les activités pédagogiques, et sur la réalisation d'entretiens individuels à la fin du parcours avec les participants au projet, leurs parents (quand ils sont disponibles) ou les professionnels qui les accompagnent. Des captations vidéo sont aussi effectuées durant les activités musicales.

Pendant la première année de la recherche (2020/2021) deux parcours pédagogiques ont été réalisés, avec deux groupes de personnes en situation de handicap mental et cognitif suivies par des associations : le premier était composé de quatre adolescents (deux filles et deux garçons) ayant le syndrome de *Williams et Beuren*. Il s'agit d'une maladie d'origine génétique, caractérisée par une déficience intellectuelle de légère à modérée, des capacités motrices et un rapport à l'espace problématiques, un comportement hyper-social et, souvent, une aptitude pour la musique et un sens du rythme assez développé. Le deuxième parcours a été réalisé avec un groupe de quatre adultes (trois hommes et une femme) ayant plusieurs types de handicap cognitif : il s'agissait principalement de déficits importants liés à la mémoire et au langage oral. Un participant avait aussi les membres inférieurs, et une partie des membres supérieurs, paralysés. Ces premières expérimentations ont permis de tester, et d'apporter des améliorations aux prototypes des applications réalisés : en effet, en cohérence avec la plupart des projets dans ce domaine, le design des instruments de musique numériques a suivi un processus itératif et participatif, dans le cadre duquel les utilisateurs ont joué un rôle central [4][20]. Les participants ont testé plusieurs types d'outils, et leurs réactions ont été observées à chaque étape du processus de développement des prototypes, comme indiqué dans les recommandations élaborées par Graham-Knight et Tzanetakis [6]. Les entretiens qualitatifs, ainsi que l'échange constant avec les parents et les professionnels des associations partenaires, ont permis d'intégrer les données collectées à travers l'observation.

3. L'IMPLEMENTATION INFORMATIQUE DES APPLICATIONS : EXEMPLES D'APPLICATIONS WEB DEVELOPPEES

3.1 L'application « La mer et les vagues »

Cette application simule les bruits de la mer et des vagues, reproduits en utilisant la technique de la synthèse soustractive et notamment en filtrant un bruit rose et un bruit blanc, une méthode classique dans le domaine de la synthèse de textures sonores [15]. Le déclenchement des vagues peut se faire avec le mouvement du dispositif, à travers le *mapping* de l'accéléromètre. En effet, l'application a été conçue pour des dispositifs portables (*smartphone*, tablette) et vise à sensibiliser l'utilisateur au geste instrumental. Pour ceux qui ne peuvent pas profiter de cette fonctionnalité, à cause de difficultés liées à la coordination motrice, il est quand même possible de déclencher les vagues en appuyant sur un bouton.

En ce qui concerne le développement de l'application, le processus de traitement du signal a été codé en Faust :

le bruit des vagues est créé par un générateur de bruit rose (généralisé en utilisant la fonction standard de Faust *no.pink noise*), filtré avec un filtre *butterworth* passe-haut (la fonction standard de Faust *fi.highpass*). Le signal est multiplié par un générateur d'enveloppe (fonction *en.adsr*) dont la *release* (*rel*), qui représente la longueur de la vague, est contrôlée par un *horizontal slider*. La fonction primitive *select2* de Faust permet de choisir le mode de déclenchement de la vague : en appuyant sur le bouton, ou en utilisant le mouvement du dispositif. Le fonctionnement de ce deuxième mode est basé sur la fonction *gate_acc*, qui lisse la valeur de l'accéléromètre (récupérée dans le code Javascript, comme expliqué après), et ensuite la transforme en une valeur comprise entre 0 et 1 pour pouvoir déclencher et arrêter le son.

```
rel = hslider ("LENGTH_WAVE", 3.5, 2.0, 6.0,
0.001) : si.smoo;
envelope = en.adsr
(0.1,0.01,0.9,rel,gate_wave)*gain;
s = hslider("SELECTOR_GATE",1,0,1,1);
gate_wave = gate_acc, gate_but : select2(s);
acc_x = hslider("XROTATION",0,-100,100,0.001);
gate_acc = +(abs(acc_x + acc_x') > 100) ~
*(0.99) : min(1) : max(0);
process = (sea_synth * gain2) / 2,
(wave_synth * envelope : echo) / 2;
```

Le bruit de la mer est simulé de manière analogue à celui de la vague, avec un générateur de bruit blanc (fonction *no.noise*) filtré par un filtre *butterworth* passe-bas (fonction *fi.lowpass*). Dans la fonction *process* les deux sources sonores sont mises en parallèle et leur amplitude est divisée par 2, pour éviter la saturation du signal. Un effet écho codé en Faust est ajouté aux vagues.

En suivant la procédure précédemment décrite, le code Faust est transformé en nœud de l'*Audio Context* et ensuite connecté à la sortie de la machine. Dans le script Javascript, un *slider* (élément HTML `<input>`) gère le contrôle de la longueur de la vague, à travers un gestionnaire d'événements associé ; les volumes de la vague et de la mer sont contrôlés de la même manière. Enfin, un bouton permet de sélectionner le mode de déclenchement de la vague : s'il est activé, la valeur du paramètre "SELECTOR_GATE" du code Faust mentionné ci-dessus, récupérée avec la méthode *getParamValue* du nœud, est égal à 0, et le démarrage par l'accéléromètre est sélectionnée ; en revanche, lorsque le bouton est désactivé, le déclenchement de la vague se fait en cliquant sur un autre bouton placé sur l'interface web.

```
const lengthwaveControl =
document.getElementById('length');
lengthwaveControl.addEventListener('input',
function() {
merapp_dsp.setParamValue("/merapp/LENGTH_WAVE",
parseFloat(event.target.value));
}, false);
val = 0;
merapp_dsp.setParamValue("/merapp/SELECTOR_GATE",
val);
```

Dans le code Javascript, la valeur de l'accéléromètre (mouvement autour de l'axe des x) est récupérée en utilisant le constructeur *Accelerometer* de la *Api Sensor* ;

la lecture est faite 60 fois par seconde. En utilisant la fonction *convertRange*, les valeurs récupérées sont ensuite converties en valeurs comprises entre -100 et 100 (le min. et max. du paramètre XROTATION dans le code Faust), et visualisées graphiquement au sein d'un *slider* (l'élément ayant comme id 'acc') présent sur l'interface.

```
function convertRange(value, r1, r2) {
    return (value-r1[0])*(r2[1]-r2[0])/(r1[1]-r1[0]) + r2[0];
}
let acl = new Accelerometer({frequency: 60});
acl.addEventListener('reading', ()=> {
    var accX = acl.x;
    var accScale = convertRange(accX, [-10,10], [-100,100]);
    merapp_dsp.setParamValue("/merapp/XROTATION", parseFloat(accScale));
    const triggerControl = document.getElementById('acc');
    triggerControl.value = accScale;
});
```

3.2 Le « Djembé séquenceur »

Il s'agit d'une application qui simule les trois types de sons principaux d'un djembé, c'est-à-dire la basse, le son tonique et le son claqué (ou *slap*). Un *step sequencer*, composé de trois lignes de boutons qui déclenchent les sons mentionnés, permet de programmer des *patterns* rythmiques, de les répéter en boucle et les modifier en temps réel. Il s'agit d'une interface accessible et intuitive, à travers laquelle les utilisateurs peuvent visualiser et comprendre assez facilement la structure interne des *patterns* rythmiques. La vitesse et les volumes des trois sons sont contrôlés par des *sliders*. J'ai développé cette application en suivant un modèle illustré sur le site *MDN Web Docs*²⁷, dont j'ai modifié l'interface, pour la rendre plus accessible, et le processus de traitement du signal²⁸. En effet, contrairement au modèle, qui est entièrement codé en Javascript, le dsp de mon application est développé en Faust, à partir de la fonction *pm.djembe* qui reproduit le timbre du djembé en utilisant la synthèse par modèles physiques. Dans le code Faust les trois sons mentionnés sont déclenchés par des boutons (à leur tour activés par les boutons présents sur l'interface web de l'application), et leur volume est contrôlé par un *hslider*. Les paramètres de chaque son (fréquence de la fondamentale, *strikeposition* et *strikessharpness*) ont été déterminés de façon empirique, de manière à pouvoir bien distinguer les trois sons lorsque l'application est utilisée sur des dispositifs (*smartphones*, ordinateurs portables) ayant des enceintes non professionnelles de petites dimensions. Dans la fonction *process* les trois sources sonores sont mises en parallèle. Le dsp est enfin transformé en nœud dans le script Javascript.

```
gatebass = button("gatebass");
```

```
gainbass = hslider("Volumebass", 0.95, 0.0, 1.0, 0.01);
bass = vgroup ("bass", gatebass : pm.djembe(70, 0.1, 0.9, 1) * gainbass);
```

Le mécanisme à la base du séquenceur a été codé en Javascript et s'appuie sur la propriété *currentTime*, qui récupère la durée en secondes du temps écoulé après la création de l'*AudioContext*. Cette propriété, qui est basée sur le *audio clock* interne de l'ordinateur/dispositif, permet de programmer donc une suite d'événements sonores synchronisée avec cet horodatage; grâce au niveau de précision très élevé de *currentTime* (la propriété restitue une valeur arrondie au 15ème chiffre décimal), cette suite d'événements sonores reste cohérente même pendant de longs intervalles de temps. Le fonctionnement du séquenceur est basé sur un système de planification qui permet de prévoir en avance à quels moments les sons seront joués; les événements sonores détectés et planifiés sont ajoutés, avec la méthode *push*, à une « file d'attente » représentée par l'*array* stocké dans la variable *notesInQueue*. Le changement de la valeur de l'attribut *aria-checked* des *pads* (boutons) du séquenceur, modifie la valeur des paramètres *gate*²⁹ du dsp, et déclenche les sons.

```
const notesInQueue = [];
function scheduleNote(beatNumber,time {
    notesInQueue.push({note:beatNumber,time:time});
    if(pads[0].querySelectorAll('button')[beatNumber].getAttribute('aria-checked') === 'true') {
        djembeseq_dsp.setParamValue("/djembeseq/bass/gatebass",1);
    } else if
        (pads[0].querySelectorAll('button')[beatNumber].getAttribute('aria-checked') === 'false') {
        djembeseq_dsp.setParamValue("/djembeseq/bass/gatebass", 0);
        // On répète la même structure pour le son tonique et claqué
    }
}
```

Chaque ligne du séquenceur est composée de 8 boutons; 1 bouton correspond à 1 *beat* donc, chaque ligne, en 4/4, équivaut à 2 mesures. Ce choix représente un point d'équilibre entre l'accessibilité de l'interface et la possibilité de créer des variations sur les *patterns* rythmiques: plus de mesures auraient permis plus de variations, mais auraient obligé à réduire la taille des boutons, en rendant l'application difficile à utiliser sur des petits écrans. A chaque *beat*, la fonction *nextNote* indiquée ci-dessous fait avancer le *timing* interne du séquenceur d'un *beat* (1 bouton), en repartant à zéro lorsqu'il arrive à 8. La variable tempo correspond à la vitesse en bpm du séquenceur par défaut, *currentNote* est le numéro de *beat* (et de bouton) initial et *nextNoteTime* l'intervalle temporel après lequel le prochain événement sonore doit avoir lieu (par défaut fixé à 0); enfin, la variable *secondsPerBeat* calcule la durée d'un *beat*.

²⁷ https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API/Advanced_techniques [consulté le 06/01/2022].

²⁸ Le *framework tone.js* comprend des classes qui peuvent être utilisées pour la création d'un séquenceur; cependant, j'ai décidé de ne

pas m'en servir et de développer le séquenceur à partir de zéro, pour améliorer mes connaissances dans le domaine de la programmation web.

²⁹ Il s'agit plus précisément des boutons *gatebass*, *gatetone* et *gateslap* du dsp, qui déclenchent les trois sons du djembé.

4. L'ANALYSE DES RESULTATS DES EXPERIMENTATIONS MENEES

```

let currentNote = 0;
let nextNoteTime = 0.0;
let tempo = 60.0;
function nextNote() {
  const secondsPerBeat = 60.0 / tempo;
  nextNoteTime += secondsPerBeat;
  currentNote++;
  if (currentNote === 8) {
    currentNote = 0;
  }
}
const lookahead = 25.0; //(millisecondes)
const scheduleAheadTime = 0.1; //(secondes)
function scheduler() {
  while (nextNoteTime < audioCtx.currentTime +
  scheduleAheadTime) {
    scheduleNote(currentNote, nextNoteTime);
    nextNote();
  }
  timerID =
window.setTimeout(scheduler, lookahead);
}

```

La variable *scheduleAheadTime* indique combien de secondes le système va vérifier à l'avance s'il y a des événements sonores programmés ; ensuite, la fonction *scheduler* vérifie si un événement sonore est prévu durant cet intervalle de temps, calculé à partir de la valeur de *currentTime*. Si tel est le cas, les fonctions *scheduleNote* et *nextNote* sont appelées. Enfin, la propriété *setTimeout* exécute la fonction *scheduler* selon la fréquence temporelle déterminée par *lookahead*. Le démarrage du séquenceur se fait à l'aide d'un bouton placé en haut de l'interface : quand il est activé, la fonction *scheduler* est appelée et la valeur initiale de *nextNote* est égale à celle relevée par la propriété *currentTime*. Le bouton déclenche aussi une autre fonction, basée sur la méthode *requestAnimationFrame*, qui permet de visualiser l'avancement du curseur du séquenceur sur les *pads*. Lorsque le bouton est désactivé, la méthode *clearTimeout* de *window.setTimeout*, et celle-ci cesse d'être appelée.

Les deux parcours pédagogiques réalisés jusqu'à présent³⁰ permettent de tracer des premières conclusions par rapport à l'impact, et au niveau de l'accessibilité, des outils développés. Ces expérimentations se sont déroulées pendant la première année de la recherche ; les dispositifs testés étaient, donc, encore à un stade non avancé : il s'agissait de prototypes développés en Max/MSP et d'applications web contenant seulement les fonctions basiques, sans la plupart des aménagements décrits au paragraphe 2.3. D'autres dispositifs comme le *Gramophone*³¹, développé par le Grame, et le capteur de mouvement *Leap Motion*³² ont été utilisés pour mettre à disposition des participants un éventail d'outils numériques et de modes d'interaction différents, afin de comprendre leurs capacités et préférences.

En ce qui concerne l'impact de ces technologies sur la réalisation du projet, trois questions ont été posées à chaque participant lors de l'entretien, visant à comprendre si les dispositifs numériques avaient facilité la réalisation et l'exécution de la composition, quel avait été le support fourni et quel type d'instruments de musique les participants auraient-ils préférés, entre les technologies numériques et les instruments de musique traditionnels. Toutes les personnes interviewées des deux groupes (six sur six) ont affirmé que les technologies ont fortement facilité la réalisation de la composition musicale ; selon deux personnes, celle-ci ne pouvait pas être réalisée avec des instruments de musique traditionnels, compte tenu du niveau des participants, qui étaient des débutants ou des personnes ayant une expérience musicale assez limitée. Cinq personnes sur six ont affirmé avoir préféré jouer avec des instruments numériques, plutôt que traditionnels. En ce qui concerne le type de support fourni par les technologies, dans deux cas la question n'a pas été posée pour des raisons liées à la longueur de l'entretien, et les autres réponses étaient pour la plupart hors sujet. Seulement une personne du deuxième groupe, présentant des déficits cognitifs et physiques assez importants, a fourni des éléments pertinents en affirmant que les instruments de musique numériques permettent de produire du son plus rapidement, et que grâce à l'ordinateur, il a pu jouer d'une seule main. Les parents des participants, présents seulement dans le cas du premier groupe, ont tous confirmé l'importance fondamentale des technologies pour atteindre le résultat final.

En ce qui concerne l'accessibilité des outils, l'observation participante a permis d'évaluer principalement deux domaines : la capacité des participants à interagir avec les applications et leurs interfaces, et la capacité à comprendre les fonctionnalités des outils et à savoir se repérer au sein des environnements numériques utilisés. En ce qui concerne le premier, les deux groupes qui ont participé aux



Figure 1. Interface de l'application.

³⁰ Voir le paragraphe 2.4.

³¹ <https://www.amstramgrame.fr/gramophone/about/>.

³² <https://www.ultraleap.com/product/leap-motion-controller/>

parcours ont montré des résultats assez analogues. En général une bonne maîtrise des outils a été constatée, et une capacité suffisante à interagir avec les éléments des interfaces. La taille des écrans a constitué un aspect névralgique. Les *smartphones*, par exemple, ont donné lieu dans certains cas à des difficultés d'interaction : les *sliders* présents au sein des applications web étaient souvent difficiles à déplacer sur les téléphones portables ; en particulier, le déplacement par étapes, comme celui d'un curseur qui sélectionne la note de l'instrument, s'est avéré particulièrement compliqué (par rapport au déplacement continu). Au cours du projet, la taille des *sliders* a été agrandie, en améliorant de manière significative l'utilisation de ces éléments. Une deuxième considération concerne le mode d'interaction, et notamment la présence d'un dispositif *hardware*, externe ou intégré avec le software : les applications qui prévoyaient ce type de dispositifs sont apparues plus attractives et intuitives par rapport à celles contrôlées à travers la souris de l'ordinateur ou l'écran tactile. Les applications commandées par des contrôleurs MIDI (il s'agit des applications développées en Max/MSP), notamment ceux caractérisés par la présence de *pads* ou de *faders*, ont été toujours choisies par les participants lorsqu'il y en avait la possibilité. En plus, l'utilisation de *faders* pour contrôler les *sliders* s'est avérée plus efficace et précise par rapport à la souris ou au déplacement du doigt sur l'écran. De façon générale, dans les cas où la dimension gestuelle est valorisée davantage, l'interaction devient plus intéressante. Cet aspect a été remarqué aussi par rapport au *Gramophone*, qui, parmi les différents dispositifs, est apparu l'un de plus attractifs pour les participants. Par contre, au-delà d'un certain seuil de complexité du geste, ce type de dispositifs peut facilement se transformer en obstacle : par exemple, avec le *Gramophone*, le réglage du volume à travers le potentiomètre rotatif a posé des problèmes, lorsqu'il devait être effectué pendant que l'appareil bougeait dans l'espace pour moduler le son en fonction de l'accéléromètre. En effet, la coordination des deux mouvements demande un niveau de dextérité supérieur à celui de la plupart des participants. Pour la même raison, l'utilisation du capteur *Leap Motion* s'est avérée complexe : le contrôle des paramètres sonores et le déclenchement des sons avec le mouvement des mains, demandent un niveau de motricité trop avancé pour les participants. Une dernière considération valable pour les deux groupes concerne la dimension visuelle de l'interaction avec les éléments de l'interface : il est très important d'avoir ce type de retour pour que l'utilisateur puisse prendre conscience des interactions effectuées. Un exemple est représenté par les boutons sur les interfaces web : en effet, tous ces éléments ne déclenchent pas du son lorsqu'ils sont pressés, et même dans le cas où cette interaction produit un retour auditif, il peut être couvert par le son des autres dispositifs utilisés dans la salle. En plus, il a été observé que le retour visuel impacte de manière positive sur la concentration de l'utilisateur. Pour ces raisons, un changement de couleur de tous les

boutons, lorsqu'ils sont pressés, a été mis en place au cours du projet.

En ce qui concerne la capacité à comprendre les fonctionnalités des outils, et à savoir se repérer au sein des environnements numériques, l'observation des deux groupes a produit des résultats différents. Les adolescents atteints du syndrome de *Williams et Beuren* ont montré une certaine aisance avec la navigation web et, plus en général, avec tous les environnements numériques utilisés. Lorsqu'ils ne comprenaient pas, ils essayaient quand même d'interagir avec les dispositifs, poussés par la curiosité. Leurs parents ont affirmé qu'ils ont l'habitude de se servir du *smartphone* et de l'ordinateur dans leur quotidienneté, pour naviguer sur internet ou utiliser des logiciels d'usage commun. La compréhension des fonctionnalités des applications a été assez immédiate, surtout en ce qui concerne les applications web qui, par rapport à celles développées en Max/MSP, disposent d'interfaces plus simples. Par contre, des difficultés ont été constatées par rapport au deuxième groupe, composé de personnes adultes ayant jusqu'à 55 ans. Bien qu'il s'agisse de personnes ayant des conditions intellectuelles normales (leurs troubles concernaient surtout la mémoire et le langage), dans certains cas même la compréhension des fonctionnalités de base des outils a posé des problèmes : par exemple, deux personnes n'arrivaient pas à trouver le bouton pour le démarrage des applications développées en Max/MSP et souvent, s'ils n'étaient pas suivis, ils s'arrêtaient devant l'écran sans effectuer aucune interaction. Un témoignage intéressant des diverses approches par rapport aux technologies numériques qui caractérisent les différentes générations, ainsi que de l'écart existant du point de vue de la maîtrise de ces dispositifs.

5. CONCLUSIONS ET PERSPECTIVES FUTURES DE LA RECHERCHE

La recherche prévoit la réalisation d'une troisième expérimentation pendant l'année 2022/2023, adressée à des élèves du collège en situation de handicap cognitif et mental. Ce public, en effet, constitue la cible principale du projet et pourra bénéficier d'une plus grande variété d'outils numériques, à un stade de développement plus avancé ; ceux-ci contiendront, en effet, tous les aménagements du point de vue de l'accessibilité indiqués dans le paragraphe 2.3. Les résultats des évaluations effectuées semblent valider le choix de mettre à disposition des participants une palette assez large d'outils, de manière à ce que chaque membre du groupe puisse trouver le dispositif, et le mode d'interaction qui lui convient le plus. Pour cela, la poursuite du travail sur les applications web sera accompagnée par la mise à jour et l'amélioration des prototypes développées en Max/MSP ; cela permettra aussi de pouvoir disposer d'outils adaptés aux différents contextes de jeu. En effet, si les applications équipées d'un contrôleur MIDI, comme celles réalisées en Max/MSP, sont plus attractives, elles sont moins adaptées lorsque l'enfant doit utiliser ces dispositifs à la maison ou à l'école, sans la

présence d'une personne experte. Au sein de ces contextes, les applications web, ubiquitaires et plus accessibles, représentent des solutions plus efficaces. Par rapport à tous ces aspects, la confrontation avec les professionnels et les parents a eu un rôle fondamental, et sera encore plus mise en valeur dans le futur.

Le système d'évaluation sera également amélioré, notamment en ce qui concerne la structuration de l'entretien et la formulation des questions ; l'évaluation portera aussi sur l'acquisition des compétences musicales théoriques, un aspect qui jusqu'à aujourd'hui a été moins approfondi, auquel sera dédiée une partie de l'observation et une série spécifique d'applications qui, comme indiqué ci-dessus, est en train d'être développée.

REFERENCES

1. Baron G.L., Depover, C., (dir.), « Impact du numérique sur les curricula et les programmes d'études », in *Les effets du numérique sur l'éducation. Regards sur une saga contemporaine*, Presses universitaires du septentrion, Villeneuve d'Ascq, 2019.
2. Burland, K., Magee, W.L., « An exploratory study of the use of electronic music technologies in clinical music therapy », *Nordic Journal of Music Therapy*, Vol. 17, 2008.
3. Denoux, S., Letz, S., Orlarey Y., Fober, D., « Composing a Web of Audio Applications », *Proceedings of the Web Audio Conference*, Paris, France, 2015.
4. Frid, E., « Accessible Digital Musical Instruments – A Review of Musical Interfaces in Inclusive Music Practice », *Multimodal Technologies and Interactions*, Vol. 3 (3), 57, 2020.
5. Frid, E., « Accessible Digital Musical Instruments: A Survey of Inclusive Instruments Presented at the NIME, SMC and ICMC Conferences », *Proceedings of the International Computer Music Conference*, Daegu, South Korea, 2018.
6. Graham-Knight, K., Tzanetakis G., « Adaptive Music Technology: History and Future Perspectives », *Proceedings of the International Conference on Computer Music*, Denton, USA, 2015.
7. Harrison, J., McPherson, A.P., « Adapting the Bass Guitar for One-Handed Playing », *Journal of New Music Research*, Vol. 46, 4, 2017.
8. Larsen, J.V., Overholt, D., Moeslund, T.B., « The Prospects of Musical Instruments for People with Physical Disabilities », *Proceedings of NIME – New Interfaces for Musical Expression*, Brisbane, Australie, 2016.
9. McPherson, A., Morreale F., Harrison J., « Musical Instruments for Novices: Comparing NIME, HCI and Crowdfunding Approaches », Holland, S., Mudd, T., Wilkie-McKenna, K., McPherson, A., Wanderley, M.M., *New Directions in Music and Human-Computer Interaction*, Springer International Publishing, Cham, Suisse, 2019.
10. Nicolas, O., « Idées d'usage des smartphones, tablettes, web et vidéo pour l'enseignement musical spécialisé », Terrien, P., Deveney, G., *L'intégration du numérique dans l'enseignement*, L'Harmattan, Paris, 2018.
11. Ren, S., Letz, S., Orlarey, Y., Michon, R., Fober, D., Buffa, M., Lebrun, J., « Using Faust DSL to Develop Custom, Sample Accurate DSP Code and Audio Plugins for the Web Browser », *Journal of the Audio Engineering Society*, Vol. 68, 2020.
12. Roads, C., *L'audionumérique : Musique et Informatique*, Dunod, Paris, 3ème édition française, 2016.
13. Samuels, K., « Enabling Creativity: Inclusive Music Interfaces and Practices », *Proceedings of the International Conference on Live Interfaces (ICLI)*, Lisbon, Portugal, 2014.
14. Schafer, R.M., *Le paysage sonore. Le monde comme musique*, Wildproject, Marseille, édition française, 2010.
15. Schwarz, D., « State of the art in sound texture synthesis », *Proceedings of the 14th International Conference on digital audio effects (Dafx-11)*, Paris, France, 2011.
16. Swingers, T., « Getting better all the time : Using music technology for learners with special needs », *Australian Journal of Music Education*, n. 2, 2009.
17. Tricot, A., *Numérique et apprentissages scolaires : quelles fonctions pédagogiques bénéficient des apports du numérique ?*, Cnesco, Paris, 2020.
18. Turner, W., *Javascript for Sound Artists: Learn to Code with the Web Audio Api*, Routledge, Londres, 2017.
19. Vamvakousis, Z., Ramirez, R., « The EyeHarp: A gaze-controlled digital musical instrument », *Frontiers in Psychology*, 7, 906, 2016.
20. Ward, A., Woodbury, L., Davis, T., « Design Considerations for Instruments for Users with Complex Needs in SEN Settings », *Proceedings of NIME - New Interfaces for Musical Expression*, Copenhagen, Danemark, 2017.