Numéros / n° 7-8 - Culture du code

Â«Â Enseigner le *patching* de manière collectiveÂ Â»

Alain Bonardi

Résumé

Les pratiques musicales collaboratives fondées sur le traitement temps réel du son à des fins de composition musicale constituent un nouvel enjeu en matière de pédagogie musicale. Notre équipe du Centre de recherche en informatique et création musicale (CICM) s?est engagée dans une recherche-action dans le cadre du projet de l?Agence nationale de la recherche Musicoll, pour essayer de mieux comprendre les enjeux des pratiques collaboratives dans le *patching*. Cette réflexion a été menée autour du développement d?un logiciel de *patching* collaboratif appelé Kiwi. Nous abordons les stratégies d?enseignement, les scénarios de pré-implémentation conçus par les auteurs du cours et leur mise en ?uvre pendant le second semestre de l?année universitaire 2017-2018. Nous traiterons du *design* pédagogique pour refondre le cours de *patching* et de l?impact attendu sur l?apprentissage et les scénarios pédagogiques mis en ?uvre. Enfin, nous aborderons l?analyse d?un cas réel constitué par une séance de test effectuée en décembre 2017 avec des étudiants de licence 3. Cette séance a permis d?éprouver la stabilité du logiciel en conditions réelles durant un cours et nous a permis de « patcher » de manière collaborative avec la classe.

1. Introduction au *patching* collaboratif

1.1. Genèse de l?article

Cet article est un compte-rendu d?étape du processus d?implémentation dans un cours universitaire du nouveau logiciel de *patching* (1) collaboratif nommé Kiwi. Il est rédigé de manière collaborative par les membres de l?équipe du CICM à l?image des méthodes de travail en cours au laboratoire.

L?équipe, se définissant volontiers elle-même comme une communauté de pratique (au sens d?Étienne Wenger, 2005) de la composition musicale, est constituée d?enseignants-chercheurs et de leurs étudiants. L?ensemble des membres de l?équipe sont des « patcheurs » utilisant les logiciels Max et/ou Pure Data et des compositeurs maîtrisant l?informatique musicale en temps réel. Quatre des auteurs ont enseigné ou enseignent l?introduction au *patching* sur ces logiciels.

Le professeur du cours, le responsable pédagogique, le *designer* pédagogique et les développeurs de Kiwi font un suivi hebdomadaire de l?enseignement du cours de *patching* sur Kiwi. Les séances débutent par des retours d?expérience mis en commun. Ces derniers alimentent les évolutions du logiciel, qui à leur tour influent sur la pédagogie (Figure 1).

Figure 1. Cycle du développement technico-pédagogique du projet Musicoll

Source : Philippe Galleron, Eric Maestri, Jean Millot, Alain Bonardi et Eliott Paris, 2018

1.2. Le *patching* collaboratif avec Kiwi

Kiwi est un logiciel de *patching* numérique dans la lignée de Max et Pure Data. Les spécifications techniques de ce logiciel du point de vue du développement ayant été publiées aux JIM 2017 (Paris, 2017), nous renvoyons le lecteur à ce texte.

Les programmes Max et Pure Data sont largement utilisés de nos jours et constituent une référence dans la pratique de la musique mixte ainsi que, par ailleurs, dans la pédagogie de l?informatique musicale. Cependant, ils ne sont pas conçus dans une optique de partage créatif et pédagogique. À l?opposé, le logiciel Kiwi que nous avons conçu au CICM dans le cadre du projet ANR Musicoll [(2)] (2015-2018) développe cette pratique en permettant l?interaction entre les programmeurs sur le même patch.

L?élaboration du logiciel Kiwi est une réponse à un double constat : le premier porte sur les limites de la modalité unique d?enseignement du *patching* et le second est relatif à la pratique du *patching* qui est jusqu?à présent plutôt individuelle et solitaire, derrière son propre ordinateur. C?est précisément sur cette tension entre un enseignement musical collectif et une pratique solitaire que le développement de ce logiciel vient agir.

L?enseignement des logiciels de *patching* est habituellement caractérisé par une approche frontale, dans laquelle la relation entre l?enseignant et la classe est de type vertical : l?enseignant transmet son savoir et sa propre manière d?utiliser les logiciels, d?une manière hiérarchique descendante. D?ailleurs, l?expérience acquise dans l?enseignement en interne à Paris 8 des environnements de *patching* confirme cette perspective. En effet, la phase d?analyse [(3)] du cours existant a mis en évidence que lors des cours de *patching*, les professeurs, chacun avec leurs propres conceptions et leurs critères, restent fidèles à ce modèle de transmission du savoir (Galleron, 2017). Cette stratégie d?enseignement exploite les caractères des environnements existants, qui favorisent ce type de transmission à cause de leur structure, laquelle ne permet pas de modalités collaboratives. En effet, si la pratique de l?enseignement du *patching* à Paris 8 se fait en classe de manière collective, la pratique du *patching* ne l?est pas à proprement parler.

Par analogie, si nous devions le comparer à une pratique collective instrumentale, le geste de programmation n?est pas directement apparent et le son produit non plus, car l?interaction visuelle et sonore n?est pas immédiate. Il faut donc un processus relativement long entre la programmation sur sa machine et l?obtention des premiers résultats sonores que l?on peut partager avec les autres. D?autre part, partager l?architecture d?un patch demande soit de transmettre le fichier d?une machine à l?autre, ce qui

demande un temps d?analyse à celui qui le découvre *a posteriori*, soit de partager un visuel sur un écran, ce qui permet d?explorer dynamiquement les ressources du patch mais n?en donne qu?un aperçu. De fait, l?interaction tant visuelle que sonore est fortement différée ; on comprend aisément alors que ce type de pratique musicale ne se prête que peu au collectif.

Or, ce manque d?interactions dans les échanges se ressent également dans l?enseignement du *patching* et a des conséquences sur l?apprentissage. En effet, dans les sciences de l?éducation, les approches socio-constructivistes inspirées des travaux de Lev Vygostky (2011) montrent l?importance des interactions dans le processus d?apprentissage ; qu?elles soient symétriques (entre étudiants) ou asymétriques c?est-à-dire de participant avancé à moins avancé (par exemple entre professeur et étudiants). C?est pourquoi le développement d?un logiciel collaboratif dans ce domaine apporte des changements dans la pratique, car elle peut se réaliser collectivement. Les délais de partage sont réduits et offrent des opportunités nouvelles en termes d?enseignement, notamment une approche de la transmission plus horizontale, qui nous conduit assez naturellement à une approche collaborative. Notre approche peut faire penser au domaine du *computer supported collaborative learning* (CSCL) qui « est une branche des sciences de l?éducation qui s?intéresse à l?étude de la façon dont les gens peuvent apprendre ensemble avec l?aide des ordinateurs [(4)] » (Stahl, Koschmann et Suthers, 2006).

Par ailleurs, notre équipe concilie des méthodes et points de vue pédagogiques complexes, à l?instar des CSCL qui considèrent des conceptions opposées de l?enseignement (*Id.*), comme :

- la métaphore de l?acquisition « dans laquelle l?apprentissage consiste en l?acquisition de connaissances stockées dans l?esprit [(5)] » ;
- la métaphore de la participation « dans laquelle l?apprentissage consiste en une participation croissante aux communautés de pratique [(6)] » ;
- la métaphore de la création des connaissances qui « sont créées [?] grâce à la collaboration [(7)] ».

Enfin, notre méthodologie de recherche comme dans les CSCL combine des approches « de conception expérimentale, descriptive et itérative [(8)] » (*Id.*).

Toutefois, nous devons bien signifier que Kiwi n?est pas, à ce stade, pensé comme un dispositif d?apprentissage en ligne, ni pensé pour faciliter l?apprentissage de la musique à l?aide d?un ordinateur. Il est enseigné et pratiqué en présentiel. Il participe à l?apprentissage de l?audionumérique, tout en apprenant le *patching* à des fins de création musicale. Pour cela, il est conçu pour développer un enseignement et une pratique musicale collaborative.

Dans cet article nous discuterons comment, au CICM, nous élaborons des stratégies d?enseignement de la programmation musicale, suite à l?introduction du logiciel musical collaboratif Kiwi dans l?offre de formation en informatique et création musicale au sein du département de musique de l?université Paris 8. Nous traiterons particulièrement du *design* pédagogique et de l?impact de ce nouveau moyen technologique sur la refonte du cours « Introduction au *patching* avec Max et Pure Data ». Pour aborder ces sujets, le texte suivant s?articule en quatre parties. Premièrement, il présente de manière succincte le projet de réalisation du logiciel de *patching* collaboratif Kiwi, mené dans le cadre du projet Musicoll ; deuxièmement, il discute le *design* pédagogique dans sa phase de pré-implémentation ; troisièmement, il propose un scénario-type ainsi que le déroulement d?une séance de cours et enfin, les auteurs font un retour d?expérience d?une séance de test logiciel, dans laquelle ils ont pu réaliser des patchs en mode collaboratif pour la première fois avec les étudiants et évaluer la robustesse de Kiwi en situation réelle.

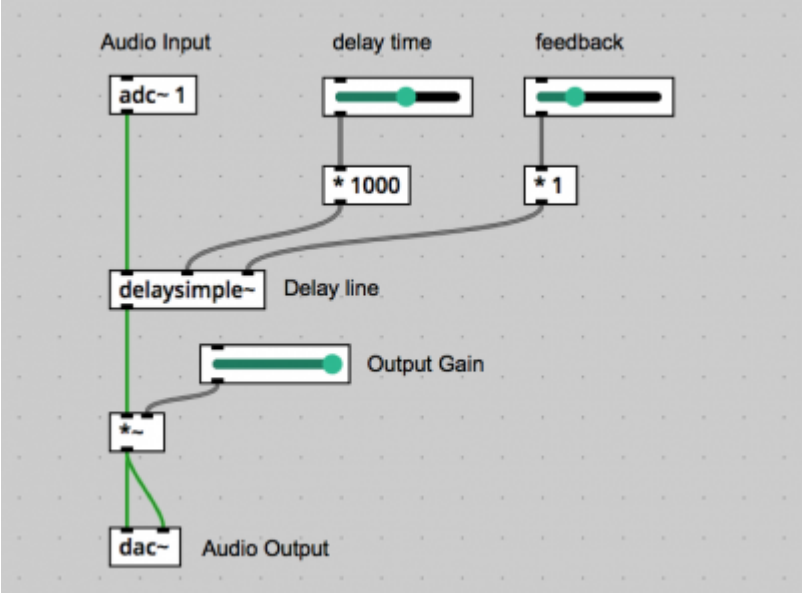
2. Le projet Musicoll : Kiwi

L?idée du développement d?un projet de recherche sur la programmation collaborative découle de l?observation des environnements de traitement du signal en temps réel existants, du constat de leurs limites en termes d?interactions collaboratives entre utilisateurs-programmeurs et de la pratique de Max et Pure Data dans le cadre de l?enseignement musical à l?université Paris 8. À partir de l?analyse du cadre de ces pratiques, les chercheurs de l?équipe-projet Musicoll ont conçu un nouveau logiciel de *patching* appelé Kiwi. Ces chercheurs souhaitent :

- développer un environnement numérique audio collaboratif pour le traitement du signal en temps réel ;
- étudier son utilisation par les musiciens-compositeurs ;
- renouveler l?enseignement des logiciels en temps réel en refondant le cours de Max et Pure Data dans le cadre de la mineure « Composition assistée par ordinateur » en licence 2 au département musique de l?université Paris 8.

Kiwi propose un environnement similaire à celui des programmes existants. La figure 2 exemplifie cet environnement. On y voit qu?à la manière de Max et Pure Data, des modules qu?on appelle « objets » sont connectés (9). L?objet **[adc~]** donne accès au signal d?entrée audio, **[delaysimple~]** correspond à un retard avec réinjection du même signal, tandis que **[dac~]** est la sortie audio ; **[*~]** permet de contrôler l?amplitude du signal et les objets **[*]** mettent à l?échelle les valeurs des curseurs pour régler les paramètres de retard et réinjection correspondants (Figure 2). À la différence des deux premiers logiciels, un tel patch peut être modifié dans Kiwi par tous les utilisateurs y ayant accès.

Figure 2. Patch Kiwi appliquant un retard avec réinjection et un gain sur un signal audio



Source : logiciel Kiwi, Philippe Galleron, Eric Maestri, Jean Millot, Alain Bonardi et Eliott Paris, 2018

3. Conceptualiser l?enseignement du *patching* collectif au moyen d?un nouveau *design* pédagogique

L?un des principaux objectifs du projet Musicoll est de repenser et renouveler l?enseignement de l?interaction musicale en temps réel au service de la composition. Les enseignants et leurs étudiants sont associés dans la conception de nouvelles stratégies pour apprendre à « patcher » de manière collaborative.

L?esprit qui guide la conception pédagogique de ce cours se fonde sur l?acquis d?expérience suivant : la qualité et la rapidité de l?apprentissage semble être favorisée par le partage des travaux et des outils logiciels par la communauté des étudiants. En effet, nous pensons que les interactions dans la classe qui peuvent être symétriques ? comme par exemple la co-résolution entre pairs ? ou asymétriques ? comme par exemple professeur-étudiants et expert-novices ? « interviennent intrinsèquement dans la mise en

?uvre des activités cognitives résolutoires et dans [la] genèse des processus intra-individuels de développement des compétences » (Roux, 2004, p. 2). La conception pédagogique que nous proposons est en cohérence avec ce propos. Il s?agit d?une approche de type socio-constructiviste considérant, dans notre cadre d?enseignement, que les interactions sociales autour de cette pratique musicale génèrent des conflits cognitifs qui sont une source d?apprentissage. Pour tirer parti de ce nouveau logiciel aux possibilités d?interactions sociales étendues, nous avons pour cela choisi de re-*designer* le cours, en visant tout d?abord à redéfinir ses objectifs, les outils et les scénarios pédagogiques pour l?enseignement de la programmation musicale temps réel.

4. Portage des objectifs pédagogiques dans un nouvel environnement

4.1. Anticipation des changements occasionnés

L?un des objectifs que nous nous sommes fixés dans cette refonte est de renforcer au sein du cours le *savoir-apprendre* et le *savoir-travailler* de manière collaborative, afin de montrer aux étudiants une manière alternative d?apprendre, de travailler et de créer. Une telle approche de l?apprentissage numérique collaboratif offre une contribution concrète à la transformation des pratiques pédagogiques (Tao, Zhang et Gao, 2017). Ces compétences sont considérées comme essentielles pour l?avenir du travail et de l?enseignement, comme le souligne l?organisation Partnership for 21st Century Learning [(10)] (P21). Par ailleurs, l?actualité de cette approche est en phase avec des projets de création récents. Grâce au développement des technologies numériques de partage des données, les pratiques musicales et artistiques évoluent fortement dans le sens du collaboratif en questionnant le statut traditionnel du compositeur, comme nous avons pu le constater directement [(11)] (Agostini *et al.*, 2017). Grâce à des ajustements collaboratifs dans leur façon de « patcher », les étudiants devraient apprendre à définir des règles de partage et aussi de *workflow* communes, tout en construisant leur environnement de programmation audionumérique et leur espace de travail, pour l?intégrer de manière cohérente à leur propre pratique musicale.

À travers le *design* pédagogique de ce nouveau cours, nous essayons d?anticiper l?impact que pourrait avoir Kiwi sur le professeur et sa stratégie d?enseignement du *patching*. D?autre part, avant la véritable mise en place qui a eu lieu à partir du deuxième semestre de l?année universitaire 2017-2018, nous avons essayé d?entrevoir les effets sur le groupe des étudiants de cette nouvelle forme d?interaction visuelle informelle qui n?existait pas dans le cours jusqu?à présent. En effet, l?enseignement du *patching* dans l?ancien cours était généralement composé de deux étapes. Tout d?abord, chaque étudiant préparait son patch de manière locale sur son ordinateur, puis le partageait sous la forme d?un fichier pouvant être dupliqué et ouvert sur d?autres machines, chaque duplicata du patch étant une version indépendante dont les modifications ultérieures étaient propres à chaque étudiant. Cette collaboration asynchrone impliquait une phase solitaire de réalisation, puis des phases de partage et de retour d?expérience multiples.

Le logiciel Kiwi introduit au sein du cours permet quant à lui une collaboration synchrone et collective en programmation musicale. Cette pratique est fondée sur un document unique partagé et hébergé par un serveur, localement ou sur internet, accessible et éditable par chaque utilisateur du réseau sans aucune autorisation des autres utilisateurs. Les étudiants sont par conséquent invités dans cette nouvelle situation à assumer pleinement une position de connaissance partagée. Ils sont de fait dans une petite communauté musicale, au sein de laquelle ils communiquent des stratégies de partage d?objets numériques et renforcent ainsi l?apprentissage réciproque dans et à travers leur utilisation. De ce fait, la migration d?un mode de programmation musicale essentiellement solitaire à un autre, collectif, dès la création du patch, implique des changements qui ont des répercussions sur la pédagogie de la création musicale. La question est alors d?enseigner et de se « former à et par la collaboration numérique » (De Lavergne et Heïd, 2013).

Dans le tableau (Tableau 1), nous soulignons les transpositions opérées entre notre expérience pédagogique de l?ancien cours et la projection que nous avons du nouveau. Puisqu?il s?agit d?une refonte du cours de Max et Pure Data, les professeurs chargés de ce cours s?appuient sur leur savoir-faire existant en matière d?enseignement du *patching* afin de le faire évoluer. Nous souhaitons donc que les étudiants relient le *patching* aux questions de composition musicale et, en outre, qu?ils développent de manière accrue leurs compétences en termes de collaboration : le partage implique des normes de politesse dans la construction du document, des contacts fréquents, de la motivation personnelle et de l?auto-apprentissage.

Tableau . Modification de l?environnement d?apprentissage des participants du cours de *patching*

Programmer avec Max ou Pure Data	**Programmer avec Kiwi**
Espace de travail numérique individuel	Espace de travail numérique partagé
Partage asynchrone (par exemple par mail ou sur un *cloud*)	Partage synchrone sur le même documen
Mode collaboratif en présentiel	Mode collaboratif en présentiel dans la c
Espace de travail visible uniquement à proximité du poste de travail	Espace de travail visible à distance
Travail sur le même patch limité à 3-4 étudiants (derrière le même ordinateur)	Travail sur le même patch (théoriquemen

4.2. Évaluation du dispositif

4.2.1. Évaluation des étudiants

Pour évaluer les étudiants, nous procédons à trois évaluations individuelles et sommatives par questionnaires sur la plateforme Moodle du cours. La première précède le cours et nous permet d?évaluer le niveau de connaissance des étudiants en audionumérique. La seconde, à mi-semestre, et la troisième à la fin du semestre portent sur le fonctionnement des objets de Kiwi et leurs connexions.

Par ailleurs, nous mettons en place un travail artistique qui nous permet d?évaluer chaque groupe. Il s?agit d?un format de type « concert », dans lequel les groupes doivent présenter une petite pièce musicale mettant en ?uvre un ou plusieurs procédés appris dans Kiwi (traitements, synthèse et transmission de données de contrôle). Les étudiants doivent ensuite exprimer à l?oral ce qui est en ?uvre dans leur patch et, à l?issue de cette démonstration, fournir ce dernier à la communauté, accompagné d?un document explicitant sa prise en main et leur intention artistique et technique.

4.2.2. Évaluation de l?impact de l?environnement collaboratif sur le cours

Concernant l?évaluation scientifique des effets de l?introduction du collaboratif dans le cours de *patching* nous procédons en deux étapes :

- des enregistrements vidéo de la classe en activité et du professeur faisant un retour sur chaque séance ;
- des entretiens filmés appelés d?« autoconfrontation [(12)] » et de « remise en situation par les traces matérielles [(13)] ».

Nous nous inspirons de la méthodologie de recherche empirique dite « du cours d?action » (Theureau, 2010). C?est une manière de favoriser l?expression des acteurs sur leur sujet et ainsi d?enrichir l?analyse de leur activité. Nous pouvons par exemple déceler chez les participants des modifications de leur savoir-être ou préciser leur implication dans l?activité de groupe. Pour réunir des traces de leur activité, les étudiants ayant donné leur accord sont filmés ; des captures vidéo de leur écran durant le cours sont collectées.

4.2.3. L?influence du design et de ses limites

Le *design* pédagogique tel que nous le pratiquons dans le projet Musicoll procède par allers-retours entre

projection et réalisation. Les propositions d?innovations pédagogiques lors de la refonte d?un cours, tant sur les méthodes que sur les contenus, doivent être validées par l?ensemble des acteurs.

À la fin du semestre, nous allons d?une part comparer ce qui était prescrit avec ce qui a été réalisé et d?autre part mesurer ce qui a été transmis. Pour cela, nous procéderons à l?analyse par la *Method for Analysing and Structuring Knowledge* [(14)], ou Mask (Ermine, 2013) du cours préparé par le professeur, ce qui nous permettra de dégager les *savoirs*, *savoir-faire* et *savoir-être* en jeu dans la mise en activité des étudiants. Puis, nous comparerons cette somme de connaissances supposées avec les résultats des évaluations des étudiants et l?analyse des entretiens d?autoconfrontation. Enfin, nous comparerons ces résultats avec ceux obtenus l?année précédente dans le cours « Introduction à la programmation avec Max et Pure Data »

Ces données d?analyse nous permettrons à terme de dresser une évaluation qualitative sur les compétences développées par les étudiants en terme *patching* et de collaboration.

5. Résultat de l?analyse du contexte d?enseignement : un premier scénario-type

Les cours de *patching* du second semestre 2018 sont conçus comme des séances d?apprentissage en présentiel, même si, à court terme, nous envisageons un modèle d?apprentissage mixte, combinant l?apprentissage en présentiel et à distance sur la plateforme pédagogique Moodle de l?université Paris 8. Le scénario exposé ici est essentiellement destiné à un cours en présentiel, mais nous incitons les étudiants à ce que ces situations d?apprentissage avec Kiwi prennent appui sur leur propre pratique de création musicale en dehors du cours. Ce travail de fond sera valorisé en classe et constituera l?un des supports de cours utilisé par le professeur.

C?est pourquoi nous suggérons donc aux étudiants de pratiquer Kiwi afin de :

- programmer régulièrement des traitements à distance sur un patch partagé connecté au serveur internet de l?application (ce travail doit se faire au minimum en binôme) ;
- faire appel en cas de blocage sur le patch à l?aide d?un tiers (l?enseignant ou un étudiant plus avancé) pour déboguer leur patch ;
- documentariser [(15)] (Galleron, Bonardi et Zacklad, 2015) leur patch, c?est-à-dire d?y ajouter des métadonnées, par exemple écrire un chapeau sur l?objet de leur patch ou constituer un petit document d?aide à sa prise en main.

6. Déroulement d?une session-type de travail collaboratif sur un patch original produit par un étudiant

La session commence avec la correction des devoirs des étudiants. À partir d?un travail réalisé par un groupe d?étudiants (par exemple une pièce pour instrument électronique avec traitements en temps réel dans Kiwi), le professeur propose une activité collaborative autour d?un problème technique ou musical, identifié comme pertinent dans le patch présenté, par exemple l?utilisation d?un **delay** ou d?une enveloppe d?amplitude. Ensuite, le professeur invite le groupe d?étudiants à produire un patch en collaboration. Les bénéfices attendus sont les suivants :

- une autonomie accrue des étudiants dans la réalisation de leur patch ;
- l?intégration des principes de la musique par ordinateur et la capacité à construire un discours autour de leur propre problématique musicale ;

- la capacité d?exprimer son approche musicale et compositionnelle avec un vocabulaire pertinent et spécifique au *patching* collaboratif ;
- l?émergence, par cette situation collective, de l?émulation favorable également à l?apprentissage individuel.

Nous projetons un déroulement de la séance de la manière suivante :

1) Une introduction de la problématique du cours par l?enseignant depuis son bureau sur le tableau, en situation frontale vis-à-vis des étudiants. L?enseignant explique les bénéfices qu?il attend du cours et exprime explicitement qu?il est attentif à la formulation des idées sous-jacentes au patch.

2) Les étudiants procèdent à une présentation en groupe de leurs travaux de *patching*. Chaque groupe partage ses progrès, ses difficultés et ses projections pour améliorer son patch. Les autres étudiants peuvent les rejoindre sur leur document Kiwi pour suivre visuellement leurs explications. Ce moment est suivi d?échanges verbaux et d?interactions sur le logiciel entre les étudiants.

3) À partir de ce patch original présenté par leurs pairs, chaque groupe connecté au réseau Kiwi crée une copie du document partagé original (par exemple, le groupe 1 crée le document 1 et le groupe 2, le document 2, etc.) et les étudiants s?approprient ainsi le patch du groupe à partir de leur propre ordinateur. L?enseignant lui se déplace de groupe en groupe et participe aux échanges verbaux et soutient la réflexion des étudiants.

4) Chaque groupe propose une solution possible pour développer le patch et il s?ensuit un moment de discussion sur cette solution. Cela conduit à une co-assignation des tâches par groupe et au sein de chaque groupe.

5) Chaque groupe joue ses patchs pour les expliquer. Pour cela, le groupe présente son patch de manière orale et visuelle (*via* Kiwi) avec les autres groupes pour discuter de leur processus de *patching* et réaliser une auto-évaluation de celui-ci. Les solutions valides sont alors fusionnées dans le patch d?origine et/ou constitueront, pour la classe, un répertoire de pratique pouvant être mobilisé immédiatement au besoin à partir du serveur.

7. Rapport du test de terrain effectué et commentaires

Ces perspectives de conception du cours et ces scénarios-types ont pu être partiellement mis à l?épreuve lors d?une séance de test en classe préparée en décembre 2017. Cette séance, prévue selon un protocole qui permettait de tester le logiciel avec un nombre élevé de connexions (jusqu?à 21 étudiants), a permis aux auteurs de se projeter dans une situation de cours réelle et de pouvoir apprécier la portée de leur conception pédagogique ainsi que de remarquer des aspects nouveaux qui seront intégrés lors de sa mise en ?uvre effective. Dans une atmosphère de découverte, tant pour les étudiants que pour les professeurs, nous avons pu envisager des pistes d?interactions entre les membres de la classe et les enseignants. Les pédagogues et les développeurs ont alors conçu cette séance en trois étapes pour tenter de mettre à l?épreuve trois types de connexion : sur un serveur local, distant et mobile. À partir de cette base, ils ont pu valider la tenue du logiciel, des connexions de réseau et du serveur et interagir avec la classe en utilisant Kiwi pour la réalisation de trois patchs. L?objectif principal de cette séance était clairement orienté vers un test technique, car la séance était intitulée « crash test Kiwi ». En outre, tester la prise en main de Kiwi par les étudiants n?a pas été proprement évoqué lors de cette séance et les critères n?ont pas été définis en équipe. Ce test technique consistait donc à évaluer :

- la montée en charge de Kiwi ;
- le comportement de Kiwi sur les différentes configurations matérielles (système d?exploitation des étudiants et type de machine) ;

- les accès au réseau disponibles en séance (réseaux universitaires et connexions mobiles) ;
- le travail sur Kiwi à partir d?un serveur distant, puis à partir d?un serveur local déployé sur l?ordinateur d?un des encadrants du « crash test ».

Chaque membre de l?équipe avait sa grille d?évaluation personnelle selon qu?il était le professeur du cours, le pédagogue et *designer* de ce nouveau cours ou l?ingénieur de recherche et développeur de Kiwi.

7.1. Point de vue du *designer* pédagogique

L?expérience de pédagogue nous a convaincus que les conditions de travail d?une classe sont déterminantes d?un point de vue pédagogique et influencent par conséquent l?apprentissage. À travers ce test technique, le *designer* pédagogique s?est ainsi intéressé à l?analyse des supports du cours et de l?environnement de travail que nous mesurons selon les ressources matérielles à disposition et leur distribution entre les acteurs. Au-delà de ces considérations propres aux ressources matérielles, nous étions particulièrement attentifs au confort des étudiants au sein de ce dispositif de cours et à travers l?utilisation du logiciel Kiwi.

7.1.1. Conception et réalisation de la séance

Nous avons cherché à déterminer si la salle de classe et les étudiants disposaient bien de l?équipement nécessaire pour développer l?axe collaboratif renforcé que nous nous sommes fixé dans la refonte du cours. Dans le cadre de cette séance de test, il fallait donc évaluer :

- la mise en réseau ou non des ordinateurs et la présence de disques durs partagés et de serveurs ;
- les systèmes d?exploitation des machines des étudiants (et leur mises à jour et autorisations) ;
- les supports de diffusion de l?information (tableau, enceintes, projecteurs) pour diffuser le patch Kiwi et les clefs USB pour installer Kiwi sur les machines ;
- le mobilier de classe et sa modularité pour créer des îlots ou des pôles de travail (chaises et tables) ;
- la connectivité des salles en termes de prises électriques, de connexion internet par Ethernet, de niveau et de stabilité du réseau Wi-Fi, de qualité de réception du réseau téléphonique et connexions internet mobiles, etc. ;
- le nombre d?ordinateurs et de connexions internet.

En séance, nous avons déroulé un protocole de test et nous avons réalisé à chaque étape un sondage à main levée auprès des étudiants après chaque mise en situation concrète de travail sur Kiwi et recueilli leurs commentaires oraux.

7.1.2. Évaluation de l?environnement technique d?apprentissage de la classe

Une fois réalisé ce premier inventaire sur les ressources existantes et utiles au travail collaboratif, nous avons interrogé les étudiants propriétaires de ce matériel sur leur disposition à le partager. En effet, partager le matériel (leur connexion internet mobile, par exemple) est lié assez naturellement à la question de la maintenance. L?aspect collaboratif numérique implique effectivement un matériel toujours en mesure de fonctionner pour assurer le travail qui en dépend. Il n?est pas envisageable d?avoir un problème de connexion à internet persistant ou récurrent, que cela vienne du réseau ou des machines. Il faut que ce problème intervienne le moins possible et qu?il soit très rapidement résolu : un internet haut-débit Wi-Fi stable est donc nécessaire pour un usage intensif des ressources en ligne et il faut en l?occurrence avoir un serveur robuste. Si l?un ou l?autre faisait défaut, il nous faudrait des solutions de

secours, notamment en s?appuyant sur le matériel des étudiants (connexion internet mobile, par exemple, ou constitution d?un serveur local sur leur machine). C?est pourquoi, par ailleurs, nous avons testé le serveur commun de Kiwi sur un internet distant et sur un réseau local.

Pour évaluer le confort des étudiants nous avons observé :

- leur comportement quand les premiers sons sont sortis de leur machine ;
- les premiers commentaires écrits et oraux qu?ils ont partagés sur leur patch commun ;
- la fluidité de leur geste sur l?interface graphique ;
- leurs difficultés éventuelles d?installation et de connexion.

7.1.3. Bilan

L?analyse des supports et de l?environnement de travail en situation a mis en relief la question de la maintenance des connexions internet. Nous suggérons d?attribuer des responsabilités et d?en distribuer la compétence entre le service informatique de l?université, les professeurs de la spécialité, les étudiants eux-mêmes à titre individuel ou de constituer des groupes d?étudiants compétents pour maintenir le serveur en cas de *crash* et supplanter une éventuelle défaillance de la connexion internet.

Par ailleurs, dans la mesure où nous souhaitons que les étudiants puissent travailler en groupe en dehors du lieu où se déroule le cours, notre test portant sur les connexions mobiles des étudiants et des professeurs en situation de cours a été éclairant. Ce test a montré qu?un téléphone mobile peut supporter environ quatre étudiants connectés sur le serveur. En revanche, le taux de *crash* est supérieur et la qualité des interactions visuelles est moins bonne qu?en utilisant les réseaux universitaires Eduroam ou Eduspot.

Les questions d?équipement en machine et en connexion internet étant encore fortement dépendantes des lieux, nous envisageons à terme de cartographier les espaces de travail des étudiants pour pratiquer le *patching* collaboratif hors-classe (salles informatiques, studios, bibliothèque universitaire, etc.) et nous nous demanderons s?ils sont adaptés à l?évolution du cours. Il sera intéressant de s?interroger également sur la mobilité de ce dispositif technique. Par exemple, déterminer si les apprenants peuvent être mobiles et se déplacer entre deux connexions pour pratiquer le *patching* collaboratif dans différents lieux ou s?ils sont au contraire contraints d?utiliser un certain type de matériel dans un lieu spécifique.

7.2. Point de vue de l?enseignant

7.2.1. Conception et réalisation de la séance

En ouverture de séance, le professeur a rappelé la nature expérimentale du cours et demandé aux étudiants leur participation active en tant que testeurs d?un nouveau logiciel. Il a également proposé aux étudiants de concevoir cette séance comme une expérimentation rigoureuse faisant partie d?un programme de recherche sur la pédagogie de la programmation musicale. Cette introduction a permis aux participants de comprendre l?importance de leur implication dans ce retour d?expérience. Après avoir installé le programme et suivi le protocole de test du logiciel, le professeur a demandé aux étudiants de commencer à réaliser des patchs collectifs. Tout d?abord, il a demandé à la classe de créer des objets Kiwi sur le *patcher*. Les participants ont pu apprécier les similitudes et les différences de la modalité de programmation Kiwi vis-à-vis de leur précédente expérience de *patching* avec Max et Pure Data. Les étudiants, divisés en groupes, ont été invités à réaliser un premier patch avec des objets simples permettant d?additionner des signaux. Les objets employés (**[osc~]**, **[+~]**, **[*~]**, **[slider]** et **[dac~]**) ont pu être créés par tous les groupes et connectés de manière habituelle. Le professeur a pu constater l?exactitude des patchs réalisés et interagir avec les étudiants.

Une fois tenue pour acquise la stabilité du logiciel et comprise la modalité de fonctionnement du cours

collaboratif, le professeur a proposé aux étudiants de réaliser un patch plus complexe avec un nombre d?objets plus élevé. Il s?agissait de réaliser ensemble un patch qui permette de créer une modulation d?amplitude en utilisant des objets comme **[line~]** et **[number]**. Le professeur a également abordé l?implémentation d?une ligne à retard avec l?objet **[delaysimple~]**. La modulation de fréquence implémentée en fin de séance a permis d?introduire l?objet **[sig~]** et de créer des connexions et des contrôles sonores plus fins. De cette manière, les étudiants ont pu revisiter leur façon de programmer en expérimentant une interface alternative, qui ouvre des perspectives nouvelles.

7.2.2. Évaluation technique et pédagogique

Le professeur a pu constater les différences apportées par l?utilisation de Kiwi. La classe est en effet obligée de co-participer à la réalisation du patch à cause de la structure même du logiciel. Elle implique une connexion multiple au même patch : les étudiants partagent véritablement le même espace de rédaction du *patcher*. Ceci les oblige à développer de nouvelles compétences de travail collaboratif, comme des normes de politesse et des manières de communiquer et d?interagir détournées, par exemple l?utilisation de l?objet **[comment]** en tant que messagerie instantannée. Une telle interaction introduit une dimension ludique qui favorise les échanges et la bonne atmosphère au sein de la classe. Ces retombées et les comportements des étudiants n?ont cependant pas surpris les pédagogues, car ils étaient pris en compte et prévus lors du *design* du cours.

7.2.3. Bilan

Le scénario prévu se révèle efficace pour prendre en main le logiciel. Le professeur a pu remarquer une participation plus efficace. En même temps, l?interaction avec la classe a fait en sorte que des innovations du nouveau scénario prévu aux points 3) et 4) émergent, notamment au niveau de la visualisation, en classe, des patchs réalisés singulièrement par les groupes. En contrôlant que tous les groupes étaient effectivement capables d?utiliser le logiciel, certains patchs des étudiants étaient partagés sur l?écran projeté au tableau, par l?ordinateur du professeur. Cela a permis de travailler véritablement ensemble au niveau de la programmation et d?interagir avec toute la classe pour s?entraider sur la réalisation d?un patch d?un groupe particulier. Dans cette situation, le professeur pouvait solliciter la participation de la classe en posant des questions concernant la résolution du patch. Selon les auteurs, un tel aspect constitue une avancée pratique importante au niveau de l?enseignement car les étudiants, en participant d?une manière plus active, accroissent leur compréhension du logiciel et par voie de conséquence de certains fondements de la musique électronique.

7.3. Point de vue du développeur

7.3.1. Conception et réalisation de la séance

Cette séance de test à l?université avait, du point de vue du développement et de la conception, deux objectifs principaux. Le premier était d?éprouver techniquement le logiciel dans des conditions réelles en termes de latence du réseau et de nombre d?utilisateurs connectés à une session. Le second était d?observer l?utilisation du logiciel Kiwi par les étudiants et de collecter leurs remarques afin de planifier de nouveaux développements à moyen et long terme.

7.3.2. Évaluation technique et ergonomique

La robustesse et le niveau de qualité d?un logiciel sont des notions évidemment primordiales de l?ingénierie logicielle. Cependant, dans le cadre d?un projet expérimental tel que Kiwi, l?accent est mis sur la conception de nouvelles *features* collaboratives innovantes obligeant parfois le développeur à faire des compromis. C?est pourquoi il était important de planifier une séance de test afin de relever les problèmes techniques et de pouvoir les corriger dans la phase finale précédent la sortie de la première version de Kiwi.

Comme évoqué précédemment, la qualité de l?apprentissage de l?audio numérique collaboratif est un point central dans le cadre de recherche du projet Musicoll. De plus, un de nos objectifs est de rendre accessible le *patching* audio collaboratif à une communauté large d?utilisateurs, allant du novice souhaitant découvrir cette façon de composer à l?utilisateur expérimenté. C?est pourquoi l?ergonomie et l?expérience utilisateur doivent être particulièrement soignées. Or, il est parfois difficile pour les membres de l?équipe Musicoll, qui sont soit des utilisateurs aguerris d?outils de *patching* audio tel que Max ou Pure Data, soit des personnes à l?aise avec l?utilisation de logiciels collaboratifs en réseau, de distinguer certaines difficultés d?utilisation.

7.3.3. Bilan

Du point de vue technique, cette expérience est une réussite car elle a permis d?observer des problèmes du logiciel difficiles à déceler dans un contexte classique d?utilisation, à savoir sans latence et avec seulement quelques utilisateurs. Ces problèmes seront bien entendu corrigés avant le lancement officiel du cours utilisant Kiwi comme support.

Nous avons pu constater grâce à ce test d?utilisation qu?une attention particulière devait être portée à l?ergonomie du logiciel liée à la connexion (notification de l?état connecté/non connecté, facilité du *login*).

La discussion avec les étudiants nous a également permis de distinguer des pistes de conception à plus long terme. Ainsi, la question du jeu collaboratif, c?est-à-dire de la possibilité de pouvoir transmettre entre utilisateurs des paramètres de contrôle, a de nouveau été évoquée. Nous avons donc décidé de concrétiser le développement de cette fonctionnalité afin que les utilisateurs de Kiwi puissent en bénéficier dès la première version [(16)].

Conclusions

Le changement concret d?utilisation d?un environnement de programmation graphique implique une innovation dans les approches pédagogiques qui ont un impact sur la pratique musicale en elle-même et comportent des avantages en termes d?apprentissage. Du point de vue de la conception pédagogique du cours, l?approche liée à Kiwi nous enjoint de reconstruire le lien qui associe la question de l?enseignement du *patching* audionumérique aux problématiques de composition que rencontrent les étudiants en création musicale dans notre université. Selon nous, le *patching* collaboratif apporte des innovations qui conduisent à :

- expérimenter l?intelligence collective dans ce type de pratique de programmation musicale ;
- penser les étudiants en terme de membres d?une communauté d?apprentissage ;
- repenser la façon d?évaluer les étudiants au sein d?un groupe ;
- commencer une réflexion/action au sein de l?équipe pédagogique sur l?apport de l?apprentissage en collaboratif numérique ;
- accepter de nous investir en tant qu?enseignants-chercheurs sur cette façon collaborative d?enseigner et de pratiquer la programmation musicale, avec l?intuition qu?elle sera très développée à l?avenir.

Lors des tests effectués, nous pouvons avancer des considérations concernant ce paradigme collaboratif :

- nous avons remarqué une **implication différente** de la classe qui est amenée à dialoguer d?une manière profitable dans la réalisation des patchs ;
- l?interaction entre les étudiants assume une **dimension ludique** qui crée une atmosphère favorable. Le professeur a pu apprécier des aspects nouveaux au niveau de la conduite du cours ;

- le partage des différents patchs des groupes permet d?impliquer les membres de la classe dans la résolution des questions liées à la programmation. Cet aspect a selon nous un impact remarquable sur la vitesse d?appréhension des concepts et des objets, leur profondeur et leur enracinement dans la pratique des étudiants.

Des aspects critiques accompagnent les éléments positifs de l?utilisation de Kiwi que nous venons d?évoquer :

- l?**instabilité des connexions** internet pourrait mettre en danger le déroulement des séances ;
- la mise en commun du support pourrait mener à des **déséquilibres** au niveau de la participation des étudiants. En effet, le fait de travailler sur un même patch risque de favoriser les étudiants plus rapides et/ou aguerris. C?est pourquoi, afin de remédier à cette éventualité, les enseignants ont demandé aux étudiants inscrits de remplir un questionnaire concernant leurs connaissances et pratiques audionumériques. Au cours du semestre, ils veillent à la constitution de groupes d?environ 5-6 étudiants pour favoriser la proximité entre participants de niveau équilibré dans leur aptitudes (à modéliser, à mobiliser des connaissances, à expliquer leur démarches).

De fait, l?équipe pédagogique espère qu?à travers ce dispositif d?enseignement, les étudiants auront acquis une base de connaissances qui devrait leur permettre de terminer leur licence de manière efficace et d?envisager la poursuite de leurs études en ayant profité des innovations au niveau pédagogique que ce nouveau logiciel propose. Pour mesurer le premier impact des scénarios pédagogiques envisagés, les enseignants préparent un bilan de fin de semestre en comparant les connaissances de la classe au début et à la fin du cours de Kiwi.

Remerciement

La majeure partie de la recherche exposée dans cet article est financée dans le cadre de la convention attributive d?aide de l?Agence nationale de la recherche nº ANR-15-CE38-0006-01.

---

1. Programmation graphique musicale. Le terme *patching* décrit l?action de programmation de traitements sonores dans des environnements graphiques comme Kiwi ; de manière très simplifiée, *il s?agit de connecter des boîtes* représentant des traitements avec des câbles virtuels.

2. Musicoll : Musique temps réel collaborative et nomade, projet ANR (2016-2018) associant le CICM et l?entreprise Ohm Force.

3. Selon les phases du modèle d?*instructional development Addie* (*analysis*, *design*, *development*, *implementation*, *evaluation*).

4. Notre traduisons : « branch of the learning sciences concerned with studying how people can learn together with the help of computers »

5. Notre traduisons : « in which learning consists of individuals acquiring knowledge stored in their minds »

6. Notre traduisons : « in which learning consists of increasing participation in communities of practice »

7. Notre traduisons : « in which new knowledge objects or social practices are created [?] through collaboration »

8. « Research methodology in CSCL is largely trichotomized between experimental, descriptive and

iterative design approaches. Although sometimes combined within a single research project, the methodologies are even then typically kept separate in companion studies or separate analyses of a single study. Different researchers sometimes wear different hats on the same project, representing different research interests and methodologies »

9. Par convention, les objets sont écrits entre crochets ; le « ~ » signifie un objet qui traite un signal audionumérique, par opposition à ceux qui traitent les données de contrôle.

10. *Partnership for 21st Century Learning*. URL : http://www.p21.org/component/content/article/2224 [consulté le 21/04/2020]

11. L?un des auteurs du présent article, Eric Maestri, a pu expérimenter un projet de composition collective avec le collectif de compositeurs italiens */nu/thing*, dont une grande partie s?est déroulée à distance grâce à ces outils.

12. Concrètement, le participant est filmé en commentant les vidéos de lui-même en pleine activité.

13. Le participant est amené à s?exprimer à partir des documents de travail qu?il a générés (par exemple : patchs, pièces).

14. La méthode Mask est une méthode de capitalisation, d?analyse et de structuration des connaissances.

15. Documentariser, selon Manuel Zacklad, consiste à doter les documents « d?attributs spécifiques permettant de faciliter (i) leur gestion parmi d?autres supports, (ii) leur manipulation physique, condition d?une navigation sémantique à l?intérieur du contenu sémiotique et enfin, (iii) l?orientation des récepteurs » (Zacklad, 2005, p. 11)

16. L?objet **[hub]** a depuis été créé pour cela.

Pour citer ce document:
Alain Bonardi, Â« Enseigner le *patching* de manière collective Â», *RFIM* [En ligne], Numéros, n° 7-8 - Culture du code, Mis Ã jour le 04/01/2021
URL: http://revues.mshparisnord.org/rfim/index.php?id=632
Cet article est mis Ã disposition sous contrat Creative Commons