Conception et réalisation de l’architecture d’une application informatique capable de piloter via MIDI un logiciel commercial de mix vidéo

Stage du 12/04/2010 au 18/06/2010

Responsables entreprise: Amélie CORDIER, Chantal CAPELLI, Jean-Pierre MAILLET
Responsable enseignant: Gille RAVEL

Centre de formation:
Université Claude Bernard Lyon 1 – IUT B – GEii
Remerciements

Tout d’abord je tiens à remercier François JOURDAN, qui m’a mis en contact avec le LIRIS.

Je voudrais aussi remercier Amélie CORDIER, Chantal CAPELLI et Jean-Pierre MAILLET pour m’avoir proposé ce stage et si bien accueilli.

Enfin, je voudrais remercier les enseignants de la formation GEii qui ont su m’apporter les notions indispensables à la réussite de mon stage, et plus particulièrement Mme PEAU et Mme DELLEAU pour m’avoir soutenu pendant ma recherche d’entreprise.
Résumé

Les nouvelles technologies ont fait naître de nouvelles formes d’art, et parmi celles-ci le mix vidéo. Des logiciels permettant de mixer différentes sources vidéo aussi facilement que pour la musique sont apparus depuis quelques années. Des tables de contrôle MIDI permettant de piloter ces logiciels plus rapidement et plus intuitivement permettent désormais de remplacer la souris et le clavier. Mais il reste encore difficile pour l’utilisateur de manipuler suffisamment rapidement et efficacement toutes les possibilités offertes par les logiciels d’aujourd’hui. Chantal et Jean-Pierre d’AliceA utilisent GrandVJ, un logiciel de mix vidéo, et ressentent les contraintes des tables de contrôle MIDI. C’est pourquoi ils souhaitent obtenir l’« Improjecteur » (nom donné par Chantal et Jean-Pierre mélangeant Improvisation et Projection), un logiciel qui les aidera dans leur démarche artistique et qui leur permettra de contrôler plus intuitivement GrandVJ.

J’ai été le premier à travailler sur le projet de l’Improjecteur. La première étape était donc de vérifier la faisabilité du projet, le problème principal étant de pouvoir envoyer des informations MIDI interprétables par GrandVJ. Après des tests concluants, une phase de conception a été nécessaire afin de déterminer les besoins du projet et l’architecture du programme. Enfin, la troisième phase fut la réalisation. J’ai utilisé des librairies existantes pour l’envoi de messages MIDI et la lecture de fichiers XML qui étaient nécessaire au projet. Ayant retenu le C++ comme langage de programmation, il m’a fallu apprendre le concept de programmation orientée objet que je ne connaissais pas encore. Puis j’ai transformé le code écrit en C préalablement.

Sommaire

Introduction ........................................................................................................................................... 6

I. Présentation du contexte ...................................................................................................................... 7

1. Le LIRIS ................................................................................................................................................ 7
   a. Historique ......................................................................................................................................... 7
   b. Le laboratoire ................................................................................................................................... 8
   c. L’équipe SILEX ................................................................................................................................. 8

2. AliceA ................................................................................................................................................. 9

3. Environnement de stage ..................................................................................................................... 10

II. Rapport technique ............................................................................................................................. 12

1. Présentation de GrandVJ .................................................................................................................... 12

2. Étude de faisabilité ............................................................................................................................. 13

3. Conception ........................................................................................................................................ 14
   a. Module d’instructions ...................................................................................................................... 15
   b. Module d’import des données GrandVJ ......................................................................................... 15
   c. Module d’automapping ................................................................................................................... 15

4. Choix techniques et environnement de travail .................................................................................. 17
   a. Déterminer le langage et choisir une librairie MIDI ....................................................................... 17
   b. Choisir une librairie XML ........................................................................................................... 18
   c. Choisir un environnement de développement ............................................................................. 18

5. Mise en œuvre et implémentation ...................................................................................................... 19
   a. Scénario applicatif classique ......................................................................................................... 19
   b. Scénario de l’Improjecteur ............................................................................................................ 20
   c. Fiche technique ............................................................................................................................... 20

6. Évolutions .......................................................................................................................................... 21

Conclusion .............................................................................................................................................. 22

Bibliographie/Sitographie ...................................................................................................................... 23
Introduction

Le milieu artistique est en plein essor dans le domaine logiciel depuis une dizaine d’années grâce aux nouvelles technologies et à l’augmentation de la puissance des ordinateurs. Il est de plus en plus facile et accessible pour le grand public de manipuler l’audio et la vidéo. Le MIDI, une norme apparue dans les années 80 servant à l’échange d’informations entre les synthétiseurs, est toujours utilisé dans les récents logiciels mais est désormais encapsulé par l’USB ou encore l’Ethernet, qui lui procurent de nouvelles possibilités.

Etant moi-même impliqué dans le milieu artistique et intéressé par l’informatique et les nouvelles technologies, c’est naturellement que j’ai choisi de faire mon stage de fin de DUT avec AliceA (description plus bas) qui me proposait de créer une application permettant de contrôler GrandVJ, un logiciel commercial de mix vidéo temps réel utilisé par Chantal et Jean-Pierre pour réaliser des spectacles d’improvisation.

La technologie actuelle permet énormément de possibilités et les limites sont désormais humaines. En effet, bien que des tables de contrôle remplaçant la souris et le clavier permettent à l’utilisateur de manipuler plusieurs choses simultanément plus facilement, il reste difficile de réaliser un nombre important d’actions à une cadence élevée, l’Homme n’ayant que deux mains. C’est pourquoi un logiciel aidant (ou même remplaçant entièrement) l’utilisateur est une idée qui a du sens. Le projet n’a pas pour prétention de créer une machine douée d’un sens de l’art et du beau, mais un outil rapide capable de suivre des contraintes, d’appliquer des modèles et de les faire évoluer. Nous verrons par ailleurs les perspectives du projet détaillées dans la fin de ce rapport.

La figure 1 ci-dessous montre le fonctionnement global du système : l’utilisateur contrôle GrandVJ soit directement, soit par une table de contrôle MIDI, soit à l’aide de l’Improjecteur, et GrandVJ compose la vidéo en fonction des différents éléments qu’il reçoit.

Pour comprendre le déroulement de mon stage, nous verrons d’abord le contexte qui a permis de réaliser ce projet, puis dans un second temps, une présentation technique retraçant les étapes de mon travail. Pour conclure, nous verrons les perspectives de l’Improjecteur.

Christophe JOURDAN
I. Présentation du contexte

L’équipe SILEX du LIRIS et AliceA travaillent ensemble sur le projet de l’Improjecteur. AliceA fournit le matériel, le savoir-faire artistique et ses besoins. Les chercheurs eux y voient un moyen intéressant de tester leurs méthodes d’intelligence artificielle tout en apportant des solutions aux problèmes d’AliceA. Pour moi, c’est un sujet de stage intéressant qui me permet d’appliquer mes acquis de l’IUT.

1. Le LIRIS

a. Historique

Le Laboratoire d’InfoRmatique en Image et Systèmes d’information (LIRIS), a été créé en 2003 par le regroupement de plusieurs laboratoires de recherche lyonnais travaillant dans le domaine des Sciences et Techniques de l’Information et de la Communication. Le LIRIS se répartit en cinq tutelles (ci-après figure2) : l’INSA de Lyon, l’Université Claude Bernard Lyon 1, l’Ecole Centrale de Lyon, l’Université Lumière Lyon 2 et le CNRS. Le LIRIS dispose aussi de sites à La Doua, Ecully et Bron.

Figure 2 - Les quatre tutelles du LIRIS

Christophe JOURDAN
b. Le laboratoire

Le LIRIS comprend environ 280 personnes, dont près de 110 chercheurs et enseignants-chercheur. Les activités du laboratoire se partagent en deux départements thématiques : "Image" et "Données, Connaissances, Services". Le LIRIS réalise une activité de recherche de fond sur ces deux thèmes, tout en développant un savoir-faire dans des domaines applicatifs pour la société : la culture et le patrimoine, l'intelligence ambiante, la biologie et la santé, l’apprentissage humain, les loisirs numériques. Le LIRIS dispose de cinq équipes dans chacun des deux départements, dont l’équipe SILEX faisant partie du département « Données, Connaissances, Services ».

c. L’équipe SILEX

2. AliceA

AliceA est une association loi 1901 qui a pour vocation la création artistique visuelle, dont la création de spectacles intégrant la réalisation de vidéo en direct. Cette association réunit des personnes venues de la télévision, de la musique et de l’industrie du spectacle.

AliceA a notamment réalisé « Lithographies Digitales », une collection de films graphiques mélangeant divers supports numériques pour créer des paysages libres d’interprétations. Ces films sont réalisés sans trames narratives pour être regardés à la manière d’une peinture ou d’une sculpture et d’amener le spectateur à se créer sa propre interprétation.

AliceA dispose d’un site web sur lequel on peut regarder certaines de leurs réalisations : http://www.asso-aicea.fr/
3. Environnement de stage

J’ai effectué mon stage à la tutelle du LIRIS se trouvant sur le site de l’Université Claude Bernard Lyon 1 (Figure 4). Le laboratoire se trouve au deuxième étage du bâtiment Nautibus (Figure 5). Je disposais d’une place dans le bureau d’Amélie Cordier où j’avais pour seul matériel un ordinateur portable et une table de contrôle MIDI tout deux fournis par AliceA (Figure 6).
Figure 5 - Bâtiment Nautibus

Figure 6 - Mon poste de travail
II. Rapport technique

1. Présentation de GrandVJ

GrandVJ (VJ pour Video Jockey), de la firme ARKAOS, est un logiciel de mix vidéo temps réel disponible dans le commerce et utilisé par Chantal et Jean-Pierre. Il permet de mixer jusqu’à 8 vidéos/effets organisés en couches (ou calques, à la manière de Photoshop) en temps réel. Les vidéos et effets sont placés dans des cases appelées triggers, appartenant à des banques, que l’on active pour charger l’image ou l’effet dans la couche sélectionnée. Chaque effet chargé dans une couche est appliqué à toutes les couches inférieures. GrandVJ peut être contrôlé par liaison MIDI.

![Figure 7 - GrandVJ 1.02](image)

J’ai travaillé avec l’avant dernière version du logiciel (Figure 7 ci-dessus), puis j’ai disposé de la nouvelle version (1.2) vers la moitié du stage. J’ai vérifié la compatibilité de mon travail avec ce nouveau logiciel, qui n’a pas revu son fonctionnement interne mais juste ajouté des fonctionnalités tel que la gestion des vidéos HD ou l’ajout d’un « crossfader » pour mixer à la manière d’un DJ... Le programme sur lequel j’ai travaillé pourra donc fonctionner avec la version 1.2 de GrandVJ dont je compte exploiter les nouveautés.

Christophe JOURDAN
2. Étude de faisabilité

Lors de mon arrivée, rien n’avait été étudié ou programmé concernant l’application sur laquelle je devais travailler. Le cahier des charges n’était pas non plus approfondi. Il a donc fallu que je réalise une étude de faisabilité afin de déterminer si le souhait d’AliceA était réalisable et comment, le but étant de réaliser un logiciel pouvant contrôler GrandVJ.

J’ai d’abord vérifié si des applications permettant de contrôler GrandVJ n’existaient pas déjà, auquel cas il ne serait peut être pas pertinent d’en faire une nouvelle. Mais à ce jour personne ne semble avoir travaillé sur cette idée, ce qui laisse toute liberté à ce projet.

Ayant déjà des connaissances dans le milieu des logiciels artistiques, j’avais une idée sur la manière dont je pourrais communiquer avec GrandVJ. En effet, la plupart des logiciels travaillant la musique ou la vidéo sont maintenant capables de recevoir des informations MIDI, GrandVJ en fait partie. De plus, j’avais déjà connaissance du fonctionnement de la norme MIDI car au cours de ma seconde année de DUT mon projet d’étude et réalisation était la fabrication d’un contrôleur MIDI matériel, une approche différente mais basée sur les mêmes protocoles.

J’ai donc réalisé plusieurs tests à l’aide d’un contrôleur MIDI et de deux programmes gratuits, MidiTest et MappleVMC. J’ai d’abord utilisé MidiTest afin d’observer les messages MIDI que transmet le contrôleur. Ensuite, j’ai utilisé MappleVMC qui installe 4 ports MIDI virtuels (le but étant de faire communiquer deux programmes entre eux il est nécessaire de créer un « câble » virtuel par lequel circuleront les informations), et j’ai pu envoyer des trames MIDI à l’aide de MidiTest sur les ports virtuels. Les tests que j’ai réalisés m’ont permis de constater que GrandVJ reconnaissait bien les trames MIDI que je lui envoyais grâce à MidiTest, et que donc il était possible de communiquer avec GrandVJ depuis un autre programme. De plus, GrandVJ est conçu pour écouter plusieurs ports MIDI simultanément, ce qui permet à un utilisateur de le contrôler depuis une table MIDI en même temps qu’un programme par exemple.

Figure 8 - Coda Nova VMX VJ, le contrôleur midi utilisé par AliceA
3. Conception

Après avoir vérifié que le projet était réalisable, il fallait pour commencer, concevoir l’architecture du logiciel (ci-dessous Figure 9) afin de cibler tout ce qu’il faudrait réaliser. Ce projet étant amené à évoluer à long terme il fallait prévoir l’architecture dans ce sens. C’est pourquoi nous avons choisi une architecture modulaire, c'est-à-dire que les différentes parties sont englobées dans des modules pouvant être utilisés séparément. À n'importe quelle étape il est alors facile d’ajouter une fonctionnalité au programme ou d’en modifier une, sans avoir à modifier l’intégralité du programme comme dans une architecture dite structurée qui ne sépare pas les différentes fonctionnalités.

La première étape de la conception fut de créer cette architecture globale qui présente les différentes fonctionnalités du logiciel : la connexion MIDI, l’import/export d’informations liées à GrandVJ, l’interface utilisateur, ...

Ensuite chaque module a été étudié séparément pour déterminer par exemple comment récupérer les informations liées à GrandVJ et sous quel format elles pouvaient être exploitées.

J’ai notamment étudié la conception d’un module gérant des modèles de comportements qui rendrait les actions de l’Improjecteur moins aléatoires. En effet, plutôt que de générer des actions aléatoirement qui ne donnent pas forcément un rendu final de qualité, il est plus intéressant de se baser sur des modèles dont on est sûrs du résultat. Ce module vient donc lire un fichier dans lequel sont formulées des actions qui donnent un bon résultat final. L’utilisateur vient écrire ce fichier d’après ses talents artistiques et ses connaissances de GrandVJ, mais il pourrait être mis à jour par l’Improjecteur lui-même (voir les perspectives du projet à la fin de ce rapport).

Figure 9 - Architecture actuelle du programme
Ce module est encore en cours de conception et ne permet pour l’instant que de charger des groupes d’effets. Je travail sur ce module avec les nouveaux stagiaires car il faut des compétences en ingénierie des connaissances.

Puis au fur et à mesure de nouveaux modules ont été ajoutés, la plupart d’entre eux sont décrits ci-dessous.

**a. Module d’instructions**

Un travail conséquent a été fait pour le module d’instructions. Ce module utilise un fichier contenant des instructions dans une syntaxe proche de l’utilisateur qui sont ensuite interprétées par l’Improjecteur et converties en informations MIDI pour appliquer ces instructions à GrandVJ. Ainsi un utilisateur peut créer une séquence d’instructions qui seront jouées par GrandVJ (charger une image puis fondu sur la transparence, etc.). Mais l’Improjecteur lui-même est capable d’écrire ce fichier. Un programme extérieur pourrait aussi écrire ce fichier, d’où l’utilité d’un fichier externe au programme.

Le principal travail sur ce module a été de créer une syntaxe permettant toutes les éventualités d’instructions (séquentielles, simultanées, mélange des deux …) pour ne pas restreindre la créativité à une limite technique. Cette syntaxe a été formulée en XML, choix détaillé dans la partie suivante.

**b. Module d’import des données GrandVJ**

La récupération des données contenues dans les fichiers de GrandVJ fut aussi une des plus importantes parties de mon travail. Ce fichier contient toutes les informations des projets GrandVJ telles que les images chargées dans le projet, les contrôles MIDI attribués aux commandes de GrandVJ... Il était donc nécessaire de pouvoir récupérer toutes ces informations afin d’interfacer l’Improjecteur correctement à GrandVJ. Ce fichier est écrit en XML. C’est ce qui a déterminé les choix concernant les fichiers externes au programme, voir dans la partie suivante.

**c. Module d’automapping**

Enfin, la troisième partie conséquente fut l’« automapping ». GrandVJ permet à l’utilisateur d’attribuer un contrôle MIDI à beaucoup de commandes GrandVJ comme l’opacité d’une image par exemple (c’est ce qu’on appelle « mapper »), ce qui permet l’utilisation d’une table de contrôle MIDI. Les informations concernant ces attributions se
retrouvent dans le fichier de projet GrandVJ qui est utilisé par l’Improjecteur. Donc si l’utilisateur ne mappe que certaines commandes (une table n’a pas forcément autant de boutons que GrandVJ a de commandes assignables, ou l’utilisateur n’a besoin que de certaines commandes), GrandVJ n’est capable d’interpréter que ces commandes, ce qui restreint beaucoup les possibilités. J’ai donc réalisé un module qui ajoute les commandes manquantes au fichier de projet GrandVJ d’après une liste de toutes les commandes existantes (crée manuellement) en leur attribuant un contrôle MIDI sans créer de conflit avec les commandes et contrôles déjà présents. Ce module permet à l’utilisateur de manipuler GrandVJ depuis une table de contrôle, sans avoir à changer sa configuration, en même temps que l’Improjecteur qui lui, est capable d’utiliser toutes les capacités de GrandVJ.
4. Choix techniques et environnement de travail

Étant le premier à programmer pour ce projet, j’avais pour contrainte de réaliser un programme lisible et pouvant être continué par d’autres. Il a donc fallu que je choisisse le langage dans lequel j’allais programmer en m’assurant qu’il permettrait de réaliser ce projet et qu’il serait suffisamment universel pour être pris en main par les prochains programmeurs.

J’étais libre de m’organiser comme je le souhaitais, mais contraint par la dépendance des différents modules. Voyons un diagramme PERT (Project Evaluation and Review Technique) montrant la dépendance des principales tâches nécessaires à la réalisation du projet :

![Diagramme PERT du projet](image)

J’ai donc commencé par traiter la liaison MIDI, sans quoi le projet n’aurait pu exister. Par la suite il m’est arrivé de traiter plusieurs choses en parallèle pour éviter de rester bloqué. Je n’ai pas réalisé l’interface graphique car je l’ai considérée moins prioritaire face au reste des tâches.

a. Déterminer le langage et choisir une librairie MIDI

J’ai cherché des bibliothèques déjà existantes permettant l’envoi de trames MIDI, ce qui me simplifierait le travail. Il en existe pour pratiquement tous les langages et c’est donc les critères vus dans la partie conception qui m’ont amené à choisir RtMidi en C++. Cette librairie est très simple d’utilisation et fait partie d’un ensemble comprenant aussi une librairie pour le traitement audio, RtAudio, qui est l’une des futures étapes du projet. De plus, l’architecture modulaire prévue convient très bien à une programmation orientée objet, tel que permet le C++. Je n’avais jamais fait de programmation orientée objet mais je n’ai pas eu de difficulté à comprendre ce concept et à l’appliquer.
Durant la première moitié du stage, je ne me suis pas focalisé sur la programmation C++ car je ne connaissais pas encore les principes de la programmation orientée objet, je codais en C chaque module séparément. Puis après avoir suffisamment avancé les premiers modules, j’ai transformé le code, réorganisé en classes et créé des objets pour rassembler tout les modules. Ce ne fut pas une perte de temps de coder d’abord en C car la structure globale du code est restée inchangée après modification.

b. Choisir une librairie XML

La lecture de fichier XML étant nécessaire pour récupérer les informations utiles et de s’interfacer correctement avec GrandVJ, j’ai cherché des librairies déjà existantes qui permettent de manipuler les fichiers XML en C++. J’ai finalement choisi TinyXML qui est simple d’utilisation et qui permet de lire et d’écrire des fichiers XML. Ce format étant très pratique pour manipuler des données et afin d’uniformiser les différents fichiers liés au programme, nous avons décidé de tous les écrire au format XML. C’est pourquoi, par exemple, le fichier d’instruction est écrit au format XML.

c. Choisir un environnement de développement

J’ai choisi d’utiliser Code::Blocks comme environnement de développement car il est gratuit, très pratique et je l’ai déjà utilisé à l’IUT, ainsi que SVN afin de partager et sauvegarder le code et les recherches effectuées.
5. Mise en œuvre et implémentation


a. Scénario applicatif classique

1. L’utilisateur ouvre GrandVJ puis charge des images et des effets dans les différentes banques en les organisant comme il l’entend.

2. Pour commencer le mix vidéo l’utilisateur sélectionne une couche parmi les 8 couches disponibles, puis sélectionne la banque et active la case dans laquelle se trouve la vidéo qu’il veut afficher. La vidéo est alors chargée dans la couche et affichée sur la sortie.

3. Pour étoffer le mix, l’utilisateur reproduit ces trois dernières étapes en sélectionnant la couche se trouvant au dessus de la couche précédente, puis en allant charger un effet de la même manière que pour la vidéo.

4. Le résultat est satisfaisant mais l’utilisateur souhaite faire réagir l’effet en rythme avec la musique qu’il écoute. Il doit donc sélectionner la couche dans laquelle se trouve l’effet, puis aller dans le panneau de réglage des paramètres liés aux effets, et faire varier à sa guise les différents paramètres.

5. Pour faire disparaître progressivement la vidéo et terminer le mix, l’utilisateur doit faire varier le paramètre d’opacité de la couche sur lequel se trouve la vidéo chargée au début du scénario. L’effet sur la couche supérieure n’affichera rien sur la sortie car l’effet s’applique aux images des couches inférieures, mais nous l’avons fait disparaître.

Que ce soit à la souris ou à l’aide d’une table de contrôle MIDI, l’utilisateur est capable de réaliser ce scénario. La différence des deux méthodes est qu’avec la souris l’utilisateur ne peut manipuler qu’une seule chose en un instant. C’est un problème lorsque l’on souhaite faire varier deux paramètres d’un effet simultanément par exemple. Dans tout les cas l’utilisateur doit gérer chaque action.
b. Scénario de l’Improjecteur

1. L’utilisateur ouvre l’Improjecteur, puis sélectionne le projet GrandVJ qu’il a préparé au préalable en y chargeant les différentes vidéos et les effets qu’il souhaite.

2. L’utilisateur appuie sur « Play ». L’Improjecteur ouvre alors GrandVJ avec le projet sélectionné et démarre le mix en chargeant des vidéos et des effets.

3. À tout moment l’utilisateur peut définir des contraintes pour l’Improjecteur, c'est-à-dire : ne charger que des vidéos de nature, donner un rendu psychédélique, suivre le rythme de la musique... Si l’utilisateur n’en définit pas, l’Improjecteur chargera des vidéos aléatoirement mais tout en respectant ses propres modèles de comportement.

4. À tout moment l’utilisateur peut juger le rendu vidéo bon ou mauvais et l’Improjecteur en tiendra compte pour redéfinir ses modèles de comportement.

5. Pour arrêter le mix, l’utilisateur appuie sur « Stop » et l’Improjecteur réalise alors un fondu vers le noir.

Que ce soit à la souris ou à l’aide d’une table de contrôle MIDI, l’utilisateur est capable de contrôler l’Improjecteur. La différence des deux méthodes est qu’avec la souris l’utilisateur ne peut manipuler qu’une seule chose en un instant, mais l’Improjecteur permettant des commandes de haut niveau, le contrôle à la souris n’est pas un problème. L’utilisateur peut donc contrôler les commandes de haut niveau par une table de contrôle MIDI, mais il peut aussi décider de contrôler directement GrandVJ par une table de contrôle MIDI en même temps que l’Improjecteur s’il souhaite rectifier certains comportements par exemple.

c. Fiche technique

Langage de programmation : C++
Environnement de développement : Code::Blocks, SVN
Librairies utilisées : RtMidi, TinyXML
Nombre de lignes de code : environ 2000
Matériel utilisé : PC avec Windows Vista, contrôleur midi VMX VJ de CodaNova
6. Évolutions

À l’heure actuelle le programme n’est en mesure d’être présenté uniquement qu’en démonstration car il opère un fonctionnement aléatoire et non intelligent. Mais il devra être suffisamment opérationnel pour octobre, date à laquelle AliceA donne une représentation.

À la 8ème semaine de mon stage, deux étudiants en L3 d’informatique sont arrivés pour travailler sur ce projet. Ils doivent s’occuper de la gestion intelligente des banques d’images, et du comportement intelligent du programme. Il est aussi prévu que je revienne travailler sur ce projet aux mois de juillet/août afin d’avancer divers modules.

Même si le programme pourra être utilisé en octobre, ce projet est loin d’être abouti. Bien que le programme arrive déjà à communiquer avec GrandVJ, il reste encore de nombreuses idées non travaillées. Par exemple, il reste le traitement audio pour faire réagir le programme à de la musique.

Et pour la suite, pourquoi ne pas imaginer un système d’interaction permettant d’évaluer les « initiatives » du programme afin de faire évoluer ses modèles de séquences et d’obtenir un résultat toujours meilleur ? Pourquoi ne pas imaginer pouvoir contrôler le programme par un contrôleur MIDI plutôt que de venir contrôler GrandVJ directement afin d’activer des comportements plus complexes ? Nous avons de nombreuses idées qui ont besoin d’être étudiées.

Il faudra sûrement attendre une à deux années avant de voir ce projet fini, mais au final ce sera un outil inédit utile et performant (et pas que pour AliceA).
Conclusion

Je souhaite continuer mes études en licence professionnelle en alternance l’année prochaine, dans le domaine de l’informatique ou de la robotique. J’aimerais pouvoir ensuite intégrer un milieu de travail proche de la recherche plutôt que de l’industrie, choix qui se confirme après avoir passé les 10 dernières semaines au LIRIS.

Ce stage m’a permis d’approfondir mes connaissances en informatique et d’acquérir une expérience pratique supplémentaire. J’ai notamment appris les principes de la programmation orienté objet, que je n’ai pas pu intégrer en option à l’IUT.

Avoir pu travailler dans un domaine mêlant la création artistique et l’informatique m’a beaucoup intéressé et je suis donc d’autant plus heureux de continuer à travailler sur ce projet cet été.

Ce stage m’a donc été très utile sur le plan professionnel mais aussi humain, m’apportant de nouvelles connaissances et une bonne expérience.
Bibliographie/Sitographie

Sites internet :

- Site officiel du LIRIS :
  http://liris.cnrs.fr/

- Site officiel d’AliceA :
  http://www.asso-alicea.fr/

- Site officiel d’ARKAOS :
  http://www.arkaos.net/

- Présentation du diagramme PERT :

- Sites relatifs à la programmation :
  http://fr.wikipedia.org/wiki/Programmation_modulaire
  http://fr.wikipedia.org/wiki/Programmation_structurée

- Sites des librairies C++ utilisées :
  http://www.grinninglizard.com/tinyxml/

Documents papier et livres :

- Guide méthodologique pour le rapport de stage DUT GEii 2010 par Catherine PEAUD

- Programmer en C++ de Claude DELANNOY aux éditions EYROLLES

Christophe JOURDAN