INSTALLASON : UN ÉDITEUR ET GESTIONNAIRE D'ESPACES MUSICAUX NAVIGABLES

António De Sousa Dias

INSTALLASON : UN ÉDITEUR ET GESTIONNAIRE D'ESPACES MUSICAUX NAVIGABLES

António de SOUSA DIAS
CICM - Centre de Recherche Informatique et Création Musicale
(MSH ParisNord / Univ. Paris 8)
CITAR - Centro de Investigação em Ciências e Tecnologias das Artes (Universidade Católica Portuguesa)
sousa.dias@wanadoo.fr

RÉSUMÉ

Nous présentons ici *InstallaSon*, une application pour l'édition et la gestion audiovisuelles interactives des installations exigeant la projection audio-visuelle et tenant compte d'interactions entre l'audio, la vidéo et l'utilisateur. Ce prototype, composé d'un ensemble de *patchs* programmés dans Max/MSP/Jitter, représente une réponse personnelle au besoin d'obtenir un outil générique pour le prototypage d'environnements musicaux navigables, tel que le projet *Monthey04*. Puis, nous discutons les avantages et limites d'une telle approche.

1. INTRODUCTION

Notre travail de composition nous a orienté vers la conception d'œuvres privilégiant deux formes, la base de données et les espaces navigables [10]. Depuis 2006, nous avons mis en oeuvre le projet « InstallaSon - KITTy... » [15], intégré depuis novembre 2007 dans le projet *PRICES* [7].

Nous allons décrire le premier axe de ce travail en cours de développement : *InstallaSon*[1]. Notre motivation artistique réside dans le désir d'exploiter d'autres formes de déploiement musical, au-delà de la séquence unidirectionnelle linéaire, plus conforme à la situation de concert. S'il existe des exemples de musiques essayant d'échapper à ce format, comme les musiques qui explorent l'ouverture et la mobilité, l'indétermination et le hasard [3], ces exemples sont habituellement destinés à une situation de concert ou à des situations très proches où le cadrage temporel est assez précis. Nous connaissons des exemples de musiques en rupture avec ce cadre telles que les *Musiques d'ameublement* d'Erik Satie (1920). On remarque par ailleurs aujourd'hui un grand essor de la pratique de l'installation, en particulier les installations où l'aspect musical joue le rôle principal. De fait, cette voie a déjà été largement exploitée et théorisée par des compositeurs tels que Robin Minard [12], Jean-Baptiste Barrière (*Reality Checks*, depuis 1996) [2], Kaija Saariaho (*Le miroir des songes*, 2004, avec J-B Barrière) et, dans un cadre de recherche, chez des compositeurs et chercheurs tels que Sedes, Bonardi et Todoroff entre autres [13].

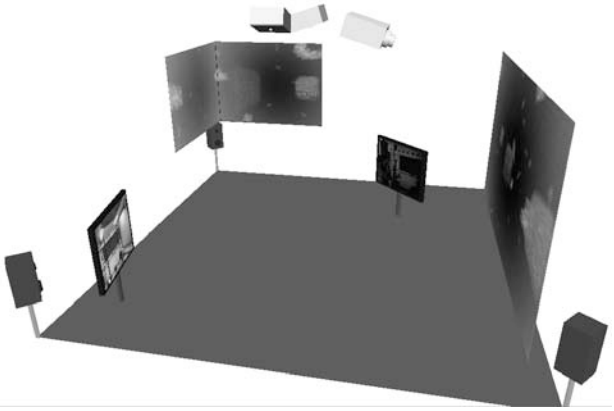

Figure 1. *InstallaSon*. Un exemple de dispositif de projection sonore et visuelle.

InstallaSon a davantage été envisagé comme un outil adapté à une situation de création où il s'agit de la conception d'une musique qui se prolonge dans l'espace (intégrant des aspects comme la mise en espace, la définition de trajectoires, la production sonore, etc.) afin de nous aider dans notre travail de composition. Comme point de départ, nous avons pris comme modèle général la situation d'installation et l'exploitation des espaces immersifs (ou du moins, « psychologiquement immersifs ») où il y a également un certain degré d'interaction avec le spectateur devenu à la fois public et usager, un « interacteur ». *InstallaSon* nourrit également notre réflexion car comme le souligne déjà Mariétan en 1969 :

> *Si le temps musical ne se limite plus à la cérémonie du concert, si l'espace est pris totalement en compte, le compositeur se trouve devenir l'aménageur qui devra assumer les conditions de la liberté d'écoute de l'auditeur ou de l'habitant.* [11]

Ainsi, d'une part, la musique est supposée être toujours « là » (elle ne commence pas nécessairement, elle ne finit pas obligatoirement) et son déroulement dépend fortement d'une interaction avec le spectateur. Une question se pose alors ici : comment structurer une oeuvre musicale (musico-visuelle) pour ce contexte ?

Une réponse partielle, très personnelle et surtout provisoire se donne dans les points suivants :

- Les situations ou scènes envisagées sont constituées des objets dont la distance par

[1] Une version *pré-alpha* d'*InstallaSon* est disponible sur Internet à l'adresse suivante : http://prices2.pbwiki.com/. Cette version, Max/MSP *stand-alone*, malgré la manque de certaines fonctionnalités permet néanmoins d'avoir une idée plus précise du fonctionnement de l'application.

rapport à un « interacteur » peut déclencher des actions comme, par exemple, la lecture d'un fichier son ou la mise à jour de son intensité.

- Ces situations ne doivent pas rester statiques. Cependant, pour maintenir un certain degré de repérage et de lisibilité de la scène il nous a fallu accepter l'intégration des aspects visuels (peut être un clin d'œil à la situation audiovisuelle de concert).
- Ces scènes sont prises comme des séquences temporelles où il existe des objets (des îles) ayant une existence intermittente (encore un clin d'œil, cette fois ci aux procédures de composition musicale habituelles).
- Les objets à constituer sont des composites image-son.
- Si la distance de l'« interacteur » par rapport à un objet introduit déjà une première forme d'interaction, il est souhaitable que dans le même composite le son puisse influencer l'image.
- Introduire la possibilité de programmer des changements abrupts de point de vue - point d'écoute dans la séquence comme une façon de dépasser un certain statisme (ici la référence vaut pour le « champ – contre champ » cinématographique comme forme d'écriture).

Notre idée de base est donc qu'*InstallaSon* génère une scène *3D*, un plateau où se déroule une séquence, dans laquelle un « interacteur » peut se promener. Cette scène/séquence prend le paradigme d'un environnement virtuel où des « interacteurs » accèdent à un monde à travers des écrans et des moniteurs audio, agissant en tant que fenêtres (points de vue et points d'écoute) sur ce monde. Leurs actions sont capturées au travers de capteurs ultra-sons, de *joystick* ou d'un autre moyen déterminé par le concepteur, elles commandent une promenade, selon des modes établis et adaptés à chaque projet.

De plus, un des principes de la conception d'*InstallaSon* est le modèle des scènes 3D interactives comme lieu de déroulement pour l'« interacteur » et comme espace de travail pour le compositeur. Par ailleurs, le choix des représentations 3D n'est pas un hasard. Comme le souligne Codognet :

> *« Il est alors intéressant de noter que, en passant du 2D au 3D, on perd en un certain sens de l'information. D'abord parce qu'il n'y a jamais de représentation parfaite comme une carte. Ensuite parce que la 3D implique toujours qu'il y ait des surfaces cachées, une « part maudite » (Bataille), la part du diable, qui échappe toujours à la connaissance. Il n'y a pas d'ombre sans lumières, de vie sans mort, à la manière baroque. »* [5]

La projection d'une scène 3D sur des surfaces 2D permet une ambiguïté que nous considérons comme un degré de liberté en raison des possibilités d'indicibilité. D'abord, on peut ne montrer que certaines parties de l'environnement ce qui permet de proposer (ou d'imposer) un chemin, ou un travail sur ce qui doit rester visible. D'autre part, concernant les effets perceptifs, on peut remarquer qu'agrandir un objet ou s'approcher de lui peut avoir le même effet. De même que voir une vidéo comme texture d'un objet (et donc apercevoir son emplacement dans l'espace) et voir cette même texture occupant tout le cadre d'un moniteur renvoie à des statuts différents pour la même image. La vidéo, dans le premier cas une image cadrée, un objet immergé, devient une image-cadre proposant une immersion en elle-même. C'est la possibilité d'exploiter cette ambiguïté qui nous a menés, par exemple, à entrevoir la possibilité de détacher les points d'accès du mouvement de l'« interacteur ».

Au final et pour fixer certaines limites, *InstallaSon* reste d'abord un outil pour la mise en scène et non pour la production des « matières premières », c'est-à-dire que les sons et les images employés sont d'abord générés au travers d'autres outils plus adaptés à cette tache. Ainsi, du point de vue de la mise en scène générale, *InstallaSon* tel qu'il est envisagé doit :

- être conçu comme un outil plutôt audiovisuel qu'auditif ;
- permettre de gérer des projections vidéo, soit au travers des projecteurs ou des moniteurs vidéo et des projections sonores au travers des haut-parleurs ;
- permettre le prototypage rapide d'une scène ;
- accommoder différents réglages ou dispositions pour une mise en scène donnée ;
- accepter un accès direct aux données engendrées pour qu'on puisse effectuer des manipulations locales ;
- faciliter la communication ou le transcodage avec d'autres outils.

La version en cours d'*InstallaSon* communique avec le logiciel Virtual Coreographer (VirChor) [9] bénéficiant d'un moteur de rendu visuel externe plus puissant et déjà en développement. Pour accomplir cela, *InstallaSon* produit également des *patchs* basés sur un sous-ensemble des commandes de VirChor.

2. LES ANTÉCEDENTS D'*INSTALLASON*

L'ancêtre d'*InstallaSon* est un *patch* Max/MSP [16] et [17] codé durant l'année 2000-2001 et permettant à l'utilisateur de placer des sons dans un espace à travers des abstractions.

Lors d'une promenade dans cet espace, l'approche d'un objet déclenchait son écoute. Le dispositif était stéréophonique à cause du type de sons, mais la restitution de la localisation de la source n'était pas prévue.

Le but de ce *patch* était de maîtriser l'objet LCD et de concevoir quelques prototypes d'objets (des abstractions) dont les propriétés étaient définies par l'utilisateur en tant qu'arguments. L'utilisateur était

obligé de faire de la programmation à l'intérieur du *patch*, en ajoutant des abstractions « manuellement ». Le maximum de dix arguments permis pour un *patch* imposait également une sévère limitation. Plus tard, un travail plus approfondi, articulant la gestion des objets `coll` et `poly~`, nous a conduit à prendre comme modèle la syntaxe définie pour les noeuds `audioclip` et `sound` dans des langages comme VRML [1]. Ceci nous a permis d'obtenir des structures de données plus consistantes et de dépasser les possibilités sonores de ce langage (assez faibles selon nous).

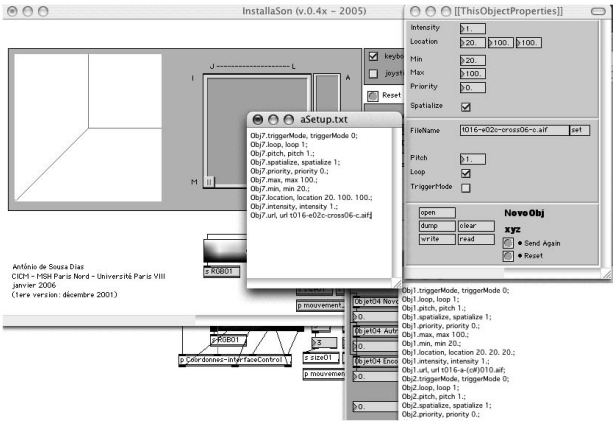

Figure 2. La version 2005 du *patch* pre-*InstallaSon*. On peut identifier l'influence de VRML si l'on examine de plus près le contenu du `coll` « aSetup.txt » aussi que le *patch* « ThisObjectProperties ».

Le désir de flexibiliser cette approche et l'inclusion dans l'environnement Max de l'objet `js` apportant des capacités de *scriptage* en *JavaScript* [8] nous a mené à un *patch* plus complexe. Le travail réalisé sur *JavaScript* dans Max nous a permis d'avoir une idée plus précise des possibilités et des limitations existantes dans la communication entre les objets Max et leur représentation *JavaScript*, mais il nous a aussi laissé entrevoir des méthodes pour dépasser ces limites. De fait,

- les variables globales *JavaScript* sont accessibles sur tout l'espace *Max*, ce qui permet une gestion plus fluide des variables et la structure de données ;
- la définition des formats de sauvegarde dans le style XML apporte la possibilité de préparer une meilleure compatibilité avec le format de sauvegarde des données sauvegardées par les objets de la famille *pattr* ; *JavaScript* permet également la communication avec cette famille d'objets.

Tout cela, nous a encore permis de dépasser des difficultés ressenties pendant la programmation d'un autre *patch* (KITTy) dans Max/MSP pour le système Mac Os 9 où un des problèmes concernait le chargement et l'installation de fichiers audio de façon dynamique en même temps que la mise en place des réglages à partir des informations chargées.

3. STRUCTURE D'*INSTALLASON*

3.1. Description générale

Comme nous l'avons dit précédemment, *InstallaSon* est composé d'un ensemble de *patchs* programmés en Max/MSP/Jitter et peut être défini comme une application pour permettre l'édition et la gestion des installations audiovisuelles interactives exigeant la projection visuelle et audio et tenant compte d'interactions entre l'audio, la vidéo et l'« interacteur » :

- Un éditeur parce qu'il permet de créer, changer, éditer et manipuler les données concernant les paramètres d'un projet ;
- Un gestionnaire parce qu'il peut gérer un projet sur place (c.-à-d. il est également responsable de l'exécution d'un projet).

Pendant l'exécution d'un projet, l'« interacteur », selon son déplacement, peut voir-écouter des objets proposés et en fonction de son emplacement spatio-temporel, il peut subir un changement abrupt de localisation par rapport à la séquence à travers d'autres objets imposant certains comportements. Le plateau où se déroule la *séquence* est accessible au travers des points de vue (des « caméras ») et des points d'écoute (des « microphones ») qui peuvent être liés ou non au mouvement de l'« interacteur ». Ces points d'accès, paramétrables par le concepteur, sont ensuite reliés à des moniteurs/projecteurs et à des haut parleurs.

Ainsi, une *séquence* consiste en la mise-en-scène des objets dans un environnement virtuel comme s'ils étaient disposés dans un « jardin ». Ces objets, formés d'autres composants, sont les briques sur lesquelles nous établissons un projet (pour le schéma global des composants d'*InstallaSon*, cf. la Figure 9). Un « jardin » est un plateau sur lequel une *séquence* choisie se déroule.

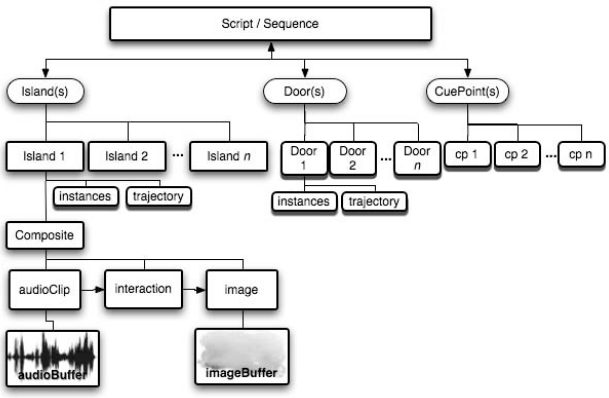

Figure 3. InstallaSon : Organisation des données dans une séquence.

Une *séquence* contient trois types d'objets (Figure 3) :

- des *îles*, objets audiovisuels accessibles à l'« interacteur » et formés des objets *composites* (des « agglomérats » hétérogènes composés d'audio et de vidéo, associés) enveloppés d'une

localisation temporelle et spatiale (cf. §3.5.1 La distribution *(Cast)*) ;

- des *portes*, lieux qui modifient le point de vue – point d'écoute dès que l'« interacteur » est dans leur zone d'influence ;
- des *points de coupe*, points temporels où le point de vue–d'écoute est obligatoirement déplacé.

3.2. Structure

InstallaSon est composé par six zones principales accessibles à travers les fenêtres suivantes :

- Les fenêtres : « installaSon », fonctionnant comme barre d'outils, et « Transport », pour déclencher une séquence choisie au préalable ;

Figure 4. Fenêtre principale d'*InstallaSon*, fenêtre *Transport* et fenêtre *Status*.

- La fenêtre pour les réglages du projet (*Project Settings*) ;
- La fenêtre de contrôle du plateau (*Stage Control*) ;
- La fenêtre du séquenceur (*Sequence Control*) ;
- La fenêtre pour gérer la distribution (*Cast*).

Du point de vue du matériel, la configuration requise suppose d'une part un ordinateur exécutant *InstallaSon*, pour recevoir les entrées d'utilisateurs et pour produire du signal audio et, d'autre part, pour chaque rendu vidéo, un ordinateur exécutant une version *client* d'*InstallaSon* (Figure 5).

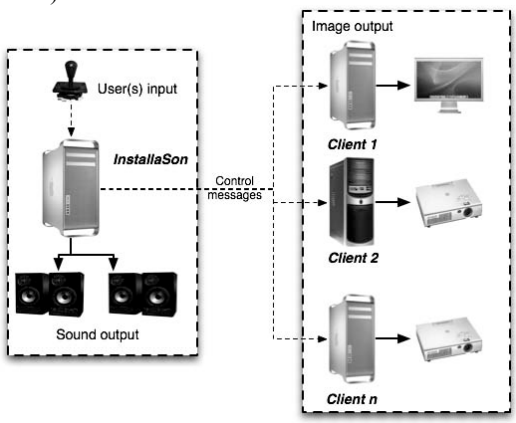

Figure 5. Exemple d'implantation matérielle.

Nous allons d'abord décrire le fonctionnement d'*InstallaSon* comme gestionnaire puis comme éditeur.

3.3. Le mode d'opération gestionnaire

Le mode de base d'opération d'*InstallaSon* comme gestionnaire d'installation peut être décrit comme suit : *InstallaSon* lit d'abord un dossier de description de projet et localise ensuite des fichiers de médias (dans un dossier nommé "lib"). Puis, selon les données d'entrée fournies par utilisateur, il produit du son et envoie des commandes aux moteurs du rendu visuel (Figure 6).

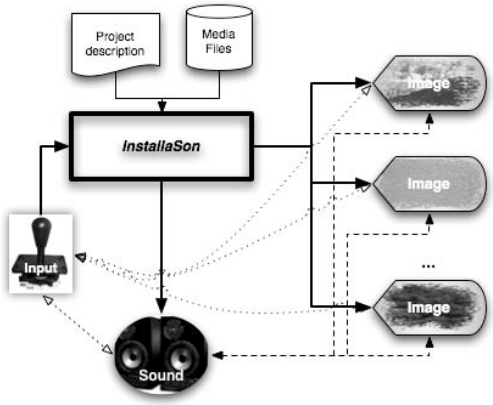

Figure 6. *InstallaSon* : Flux d'information.

Pour accomplir ceci, après le chargement d'un projet dans la machine principale, il faudra :

- charger les applications clientes appropriées dans chaque machine responsable du rendu visuel ;
- établir la communication entre machines, accomplie au travers du protocole UDP ;
- choisir la séquence à jouer (les séquences disponibles sont accessibles à travers les fenêtres « Séquenceur » ou « Transport ») ;
- déclencher la mise à jour des données des clients concernant la séquence choisie ;
- choisir le mode d'entrée / capture et configuration des données (fenêtre de contrôle du plateau) ;
- démarrer (*play*) la séquence.

3.4. La communication avec les applications de rendu d'image

Les commandes envoyées par *InstallaSon* aux applications chargées du rendu d'image relèvent de deux catégories principales :

- commandes de contrôle de la mise en place (pour écrire des fichiers contenant des descriptions de projet, etc.) ;
- commandes de contrôle de l'image pour les écrans (contrôle de caméras, mises à jour des états des objets, etc.).

Ainsi, une machine contient *InstallaSon* qui agit en tant que maître et également responsable du rendu audio

et de l'interfaçage avec l'utilisateur. Les autres machines fonctionnent en mode esclave, avec une application cliente d'*InstallaSon*. Selon les changements des paramètres, ces applications ajustent le chemin pour les fichiers contenant les médias (Figure 7).

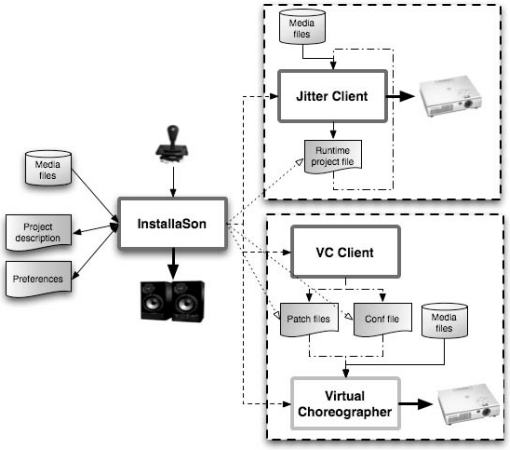

Figure 7. Les deux modes clients : le mode « client Jitter » et le mode « client VirChor ».

Il y a deux configurations possibles pour l'instant, en employant soit VirChor, soit Jitter. Dans la première, la communication est établie entre :

- *Installason* et l'application cliente afin de produire les fichiers dont VirChor a besoin : des *patch* (**.xml*) et un fichier de configuration (**.conf*) ; il produit également un fichier *batch* (*installaSonVClauncher.bat*) ;
- *Installason* et l'application VirChor.

Une fois accomplie la mise à jour du projet, il suffit de relancer VirChor au travers du fichier *installaSonVClauncher.bat*.

Dans la deuxième configuration, la communication est établie entre *InstallaSon* et l'application cliente basée en Max/MSP/Jitter produisant les *patchs* qui contiennent la description des disposition des objets ainsi que le rendu vidéo au travers d'une fenêtre *Jitter*.

3.5. Le mode d'édition

Dans *InstallaSon,* la fenêtre de dialogue de création d'un nouveau projet propose deux formes de génération : un projet vide ou un projet obtenu par génération algorithmique (à travers un *script Javascript* fourni dans le *package*, le fichier *installaSon_projectGenerator.js*). Nous faisons remarquer que ce dernier mode permet alors une entrée dans le code d'*InstallaSon*.

Lors de la génération d'un nouveau projet, *InstallaSon* met à disposition deux types d'objets :

- Des objets accessibles à travers la fenêtre de distribution et contrôle de séquence ;
- Des objets définissant les réglages de plateau / scène et de projet, accessibles au travers des fenêtres de contrôle de plateau et de réglages de projet.

3.5.1. La distribution (Cast)

La plupart des composants d'*InstallaSon* sont accessibles à travers la fenêtre de distribution (*Cast*). Nous indiquons les composants suivants, à titre d'exemple :

- *AudioBuffer* – référence à un fichier audio ;
- *AudioClip* – paramétrage et référence à un *AudioBuffer* ;
- *ImageBuffer* – référence à un fichier image ;
- *Image* – paramétrage et référence à un composant *Image* ;
- *Interaction* – paramétrage de l'influence des données extraites de l'audio sur l'image ;
- *Composite* – paramétrage et référence à des composants *AudioClip*, *Image* et *Interaction* ;
- *Île* – instanciation d'un *Composite* enveloppé des occurrences temporelles et de la situation spatiale ;
- *Porte* – instanciation des actions (changement de séquence, déplacement sur le plateau) enveloppées des occurrences temporelles et de la situation spatiale ;
- *Séquence* – un objet hybride composé d'*îles* et de *portes*, entre la séquence musicale ou un scénario de cinéma et un découpage en raison des objets *cutPoints*.

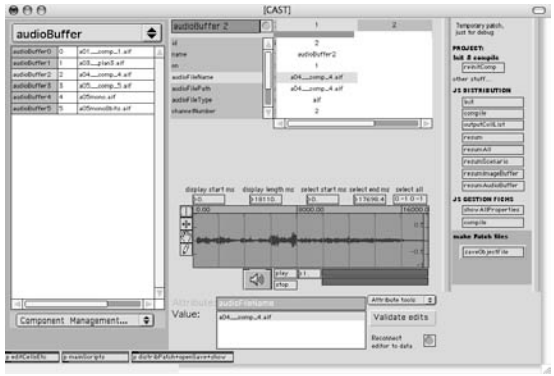

Figure 8. InstallaSon, fenêtre *Cast* : à gauche, la liste des composants *audioBuffer*, et au milieu les attributs de l'objet *audioBuffer2*.

La structure hiérarchique entre les différents composants nécessaires pour accomplir un projet *InstallaSon* est montré dans la figure suivante.

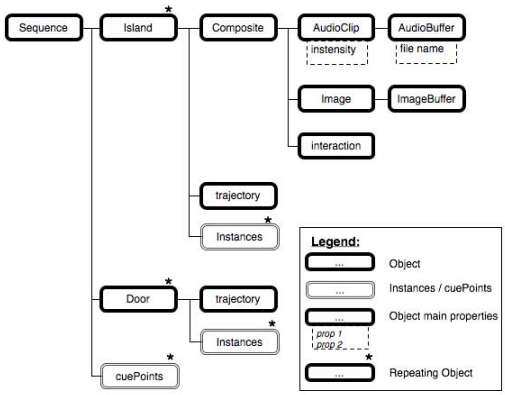

Figure 9. Composantes d'un projet *InstallaSon* et leurs dépendances immédiates.

Ces composants, malgré leur rapport hiérarchique, se présentent au même niveau du point de vue de l'édition dans la fenêtre *Cast*.

*3.5.2. Le séquenceur (*Sequence Control*)*

La fenêtre de contrôle de séquence permet de choisir, de modifier certaines propriétés et de visualiser des séquences. Ces séquences sont composées des *îles* et des *portes* placées sur un plateau virtuel. Cela signifie que celles-ci ont une position dans l'espace et une position dans le temps définie au travers des *instances*. Les *instances* sont des couples de valeurs définissant des phases de début et de fin d'apparition d'un objet sur scène. Dans *InstallaSon*, le temps est mesuré en phase. Ceci signifie que 0 (zéro) degrés représentent le début d'une séquence et 360 degrés sa fin.

Les séquences possèdent également des objets spécifiques, les *cutPoint* qui définissent des changements obligatoires de point d'écoute - de vue. *Instances* et *cutPoints* sont définis dans des objets donnés. En jargon « Orienté Objet », nous devrions les appeler des propriétés d'objet. Mais le fait qu'ils soient présents et modifiables au niveau des composants nous permet de les considérer comme des cas particuliers de composants.

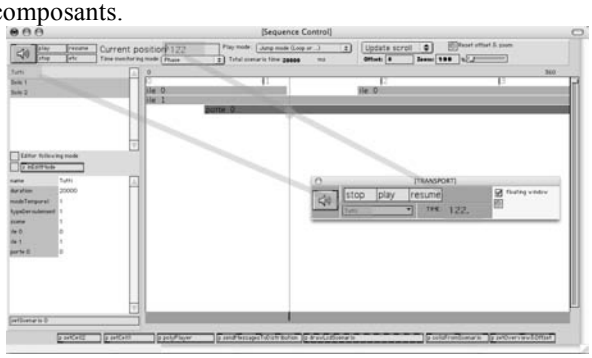

Figure 10. Fenêtre de commande de *séquence* et fenêtre de transport : dans cet exemple, la séquence contient 4 *cutPoints*, deux *îles* (la première contient deux *instances*) et une *porte*.

Le résultat est le déroulement d'une séquence sur un plateau dont les données concernant les différents objets sont articulées avec les données entrées dans le contrôle de plateau.

*3.5.3. Le contrôle de plateau (*Stage Control*)*

La fenêtre de contrôle de plateau a deux objectifs principaux :

- définir le type d'entrée des données fournies par les « interacteurs » ;
- établir l'articulation et l'envoi des données vers les moteurs de rendu vidéo.

Dans *InstallaSon*, on admet l'intégration des *patchs* programmés par le concepteur (*plug-ins*) permettant différents modes d'entrée de données : par *joystick*, par *capteurs ultra-sons*, par *trackpad*, souris, etc.

*3.5.4. Les réglages de projet (*Project Settings*)*

La fenêtre *Project Settings* permet de régler les différents paramètres généraux nécessaires pour accomplir un projet. Cette fenêtre est divisée en cinq onglets :

- fichiers et dossiers – pour choisir les dossiers utilisés par défaut ;
- réglages audio – pour paramétrer et choisir les préréglages audio, entre autres, le nombre de points d'écoute, l'assignation des points d'écoute à des haut-parleurs, leur position, et le mode de déplacement, etc.;
- réglages vidéo – paramètres concernant, entre autres, le nombre, la position, la taille et le mode de déplacement des caméras, l'attribution des points de vue à des sorties d'écran, etc. ;
- réglages pour la communication entre les différentes machines ;
- divers (pour l'instant cette fenêtre contient des réglages pour les interactions son-image).

4. ÉTAT ACTUEL DU PROJET

Notre intégration dans l'équipe du CICM à la MSH nous a permis, en 2006, de bénéficier de l'assistance de Jennifer Druart (IUT de Montreuil) pour la discussion, la définition, la structuration et la programmation de *scripts* pour la gestion, l'écriture et la récupération des données dans *InstallaSon* [6]. Entre mai et juillet 2007, des échanges entre le CICM et le LIMSI ont permis une collaboration avec Tifanie Bouchara sous forme d'un « Stage de Master 2 Professionnel Image et Son ».

Par ailleurs, l'appui de Yann Orlarey (GRAME) nous a permis de questionner et d'entamer la reformulation des aspects concernant la structuration des données y compris un approfondissement de leur signification. Par exemple, l'ancien objet *Formant* a été substitué par le *Composite* (désignation plus appropriée).

4.1. La version courante d'*InstallaSon*

La dernière version d'*InstallaSon* existe sous la forme des *stand-alone* Max/MSP/Jitter pour les systèmes Mac OS X 10.4.x (Intel) et Windows XP. Elle a un fonctionnement encore très limité permettant cependant d'entrevoir des potentialités mais ainsi que des aspects à revoir, notamment au niveau de l'ergonomie.

Une restriction réside d'ailleurs dans l'obligation de garder la même structure de dossiers, et de maintenir des copies de tous les fichiers de médias (son et image) dans les différentes machines utilisées.

Le prototype courant a désormais, entre autres, les possibilités suivantes :

- Contrôle de certaines fonctions par des menus.
- Ajout de la possibilité de la génération d'un projet par programmation de *scripts* permettant une porte d'entrée dans le code ; ceci permet la conception des projets par voie algorithmique.
- Contrôle, génération et mise à jour des fichiers externes dans différentes machines ; ces fichiers

sont des *patchs* XML ou « coll » générés automatiquement pour la lecture par les Clients.

- Addition dans le *Séquenceur* (ancien *Scénario*) de l'objet *cutPoint* ; celui-ci constitue un deuxième pas dans le rapprochement d'une écriture musicale / audio-visuelle (le premier pas étant l'existence des instances contrôlant les occurrences des îles dans le temps).
- Ajout des réglages permettant l'interférence entre les fichiers audio et image constituant un objet *Composite*. Ces réglages sont pour l'instant généraux, mais ils représentent le premier pas en direction des définitions plus détailles et individuelles pour chaque *Composite*.
- Ajout d'une fonctionnalité pour une exportation très limitée vers VRML, comprenant pour l'instant des géométries limitées à des sphères et des cubes. *InstallaSon* sauvegarde tous les séquences au format VRML (**.wrl*), une copie des fichiers son et image dans un dossier appelé "export" et ajoute une feuille html (*index.htm*) avec l'index des scénarios.

D'autres possibilités sont en cours d'implantation et nous entamons, en ce moment, une révision du code concernant les objets de base en vue d'une meilleure définition de leurs propriétés et méthodes.

4.2. Un premier résultat : *Monthey04*

Le stage de Tifanie Bouchara, « Programmation de contenus multimédias interactifs spatialisés » concernait la réalisation d'un environnement de type musée virtuel/espace navigable.

Nous avons décidé de créer comme cas d'étude, un projet artistique très limité, pour permettre d'obtenir des résultats plus précis. Le projet *Monthey04* est né de ce besoin. Il s'agit d'un prototype d'installation multimédia, à partir de l'expérience « Points d'écoute, points de vue » proposée par le compositeur Pierre Mariétan [14]. Cette expérience a été menée à Monthey (Suisse) du 28 août au 5 septembre 2004, pendant les Rencontres A.M.E.04 en association avec l'ASM. Durant cette expérience, cinq participants ont été répartis en cinq lieux. Chaque participant a passé vingt-quatre heures sur un lieu et produit un bulletin radiophonique rendant compte de ses impressions. Le lendemain, les participants échangeaient leur place de sorte qu'au bout de cinq jours, les cinq participants avaient vu les cinq lieux. Vingt-cinq interviews témoignent, entre autre, de ce travail.. Ces interviews sont à la base des vingt-cinq « îles » disposées dans un espace tridimensionnel. L'« interacteur » en se promenant à l'aide d'un *joystick* écoute les témoignages dès qu'il se situe dans leur zone d'influence.

Nous avons pris *InstallaSon* comme base pour nos travaux parallèles. Ceci a permis à Bouchara d'aboutir à des résultats précis et concrets en s'appuyant sur tous les résultats déjà obtenus concernant la recherche de ces situations, de la programmation de scènes et de la communication entre VirChor et Max/MSP. *InstallaSon* a été utilisé dans ce contexte pour la détermination et la génération de la disposition des objets, ainsi que pour étudier les interactions image-son à appliquer. Par ailleurs, l'intérêt de cette collaboration était également d'étudier les possibilités de convergence des différentes directions prises. Le point de départ d'*InstallaSon* était le son : il reste le centre de contrôle en recevant les signaux de l'utilisateur et contrôlant VirChor tandis que le résultat de ce stage est un *package*, *SceneModeler* [4] composé par des *patchs* VirChor et Max/MSP où VirChor contrôle Max/MSP.

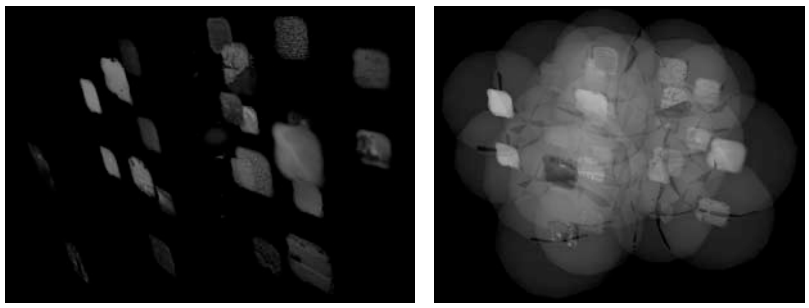

Figure 11. *Monthey04*. a) Disposition des îles b) visualisation de la zone d'influence des sons attachés à chaque île (images obtenues par exportation du projet dans le format VRML).

Figure 12. Monthey04. Rendu final dans une simulation multi écran.

5. PERSPECTIVES

La mise à disposition d'un logiciel tel que *InstallaSon* permet l'étude des cas pour une adéquation inter-plateforme, portant sur les points principaux suivants :

- Réutilisation des *patchs* et configurations déjà testés dans des projets différents.
- Utilisation de *scripts*, pour l'instant limitée à la génération algorithmique d'un projet.
- Adaptabilité, car il permet différentes configurations pour le même projet d'installation.
- Dématérialisation accomplie de deux façons : à travers la gestion des rendus audio et vidéo par des machines différentes ; à travers la déclinaison dans d'autres plateformes ou d'autres langages.

En conséquence, nous prévoyons d'aboutir prochainement à une version suffisamment stable pour mettre à disposition des versions alpha du logiciel. Un des objectifs de ce travail consiste en la mise à disposition des sources, nous proposons néanmoins pour l'instant des versions *stand-alone*, dont le mode d'emploi est plus commode.

Dans l'année en cours, nous prévoyons la mise en place des premières versions des projets ***Stillleben*** (en collaboration avec Susana de Sousa Dias) et ***Paysages*** (en collaboration avec José Xavier). Nous envisageons encore une deuxième version de ***Monthey04***, bien que, à notre avis, ce projet constitue plutôt un prototype de laboratoire qu'une véritable réalisation artistique.

6. REMERCIEMENTS

Nous remercions vivement Horacio Vaggione et Anne Sedes pour l'encadrement de ce projet au sein du CICM, ainsi que Yann Orlarey (GRAME) pour les judicieuses remarques concernant son développement. Nous remercions également Christian Jacquemin (LIMSI) pour sa disponibilité ainsi que Tifanie Bouchara et Jennifer Druart. Un remerciement à Pierre Mariétan (LAMU, AME) pour notre participation à l'expérience *Points de vue – points d'écoute*. Enfin, un remerciement à Paulo Ferreira Lopes (CITAR) pour ses relectures et commentaires très pertinents.

Ce projet est développé dans le cadre d'une bourse de post-doctorat attribuée par Fundação para a Ciência e a Tecnologia / Ministério da Ciência da Tecnologia e do Ensino Superior (Portugal).

7. RÉFÉRENCES

[1] Ames, A. L., Nadeau, D. R., Moreland, J. L. *The VRML 2.0 Sourcebook*. John Wiley & Sons, Inc., New York, 1996.

[2] Barrière, J.-B. *Installer réflexion dans le miroir électronique*, [s.d.] http://www.petals.org/Barriere/Installer.html (21/01/2008)

[3] Bayer, F. *De Schöenberg à Cage — Essai sur la notion d'espace sonore dans la musique contemporaine*. Éd. Klincksieck, Paris, 1981, p.142-189.

[4] Bouchara, T. « Le « Scene-Modeler » : des outils pour la modélisation de contenus multimédias interactifs spatialisés », 2008 (en soumission).

[5] Codognet, Ph. « Nature artificielle et artifice naturel », Buci-Glucksmann, Ch. (dir.). *L'Art à l'époque du virtuel*, L'Harmattan, Paris, 2003, p.83-94.

[6] Druart, J. *Construction d'objets pour la gestion d'environnements virtuels: rapport de stage*. CICM / IUT de Montreuil, 10 avril au 16 juin 2006.

[7] Ferreira Lopes, P. *Prices : Research Project in Interfaces and Instruments: Applications in Sonic Art*, CITAR (Universidade Católica Portuguesa), Porto, FCT ref.: PTDC/EAT/73817/2006. http://www.fct.mctes.pt/projectos/pub/2006/Painel_Result/default2.asp?idElemConcurso=911.

[8] Flanagan, D. *JavaScript – La référence* (4e édition), Éditions O'Reilly, Paris, 2002.

[9] Jacquemin, Ch, *Virtual Choreographer Reference Guide (version 1.4)*, LIMSI-CNRS and University Paris 11. http://virchor.sourceforge.net/html/index.html (21/01/2008)

[10] Manovich, L. *The Language of New Media*. The MIT Press, Cambridge (MA), 2001.

[11] Mariétan, P. *Écrit de Musique III – La musique du lieu – Musique, Architecture, Paysage, Environnement, Textes, Projets/Réalisations, Événements*. Commission nationale suisse pour l'UNESCO, Berne, 1997.

[12] Minard, R. « La musique comme espace public », Orlarey, Y. (ed.) *La Ville, Espace de Créations Sonores - Rencontres Musicales Pluridisciplinaires 2000*, GRAME, Lyon, 2000.

[13] Sedes, A. (dir.) *Espaces Sonores – Actes de recherches*. CICM, Éditions Musicales Transatlantiques, Paris, 2003.

[14] Sousa Dias, A. (org.) « Journal de l'écoute », *Sonorités – l'écoute, et après..., n.2*, Champ Social Éditions, 2008.

[15] Sousa Dias, A. *InstallaSon - KITTy : vers le développement d'outils d'assistance à la conception et à la construction d'espaces musicaux navigables – Rapport 2007*, FCT/MCTES (Portugal), nov. 2007. http://www.sousadias.com/downloads/pdf/asd_fct_Rapport2007.pdf

[16] Zicarelli, D. "An Extensible Real-Time Signal Processing Environment for Max", *Proceedings of the International Computer Music Conference*

1998. International Computer Music Association, Ann Arbor, 1998, pp463-466.

[17] Zicarelli, D., Taylor, G., Clayton, J. K., jhno, Dudas, R., Nevile, B. *MAX 4.6 : Reference Manual*. http://www.cycling74.com/

8. ANNEXE : INSTALLASON – STRUCTURE DES FICHIERS

Ci-après (Figure 13) on présente la structure hiérarchique des principaux fichiers composant *InstallaSon* et leur agencement : fichiers Max/MSP/Jitter (*.pat), fichiers JavaScript associés (*.js).

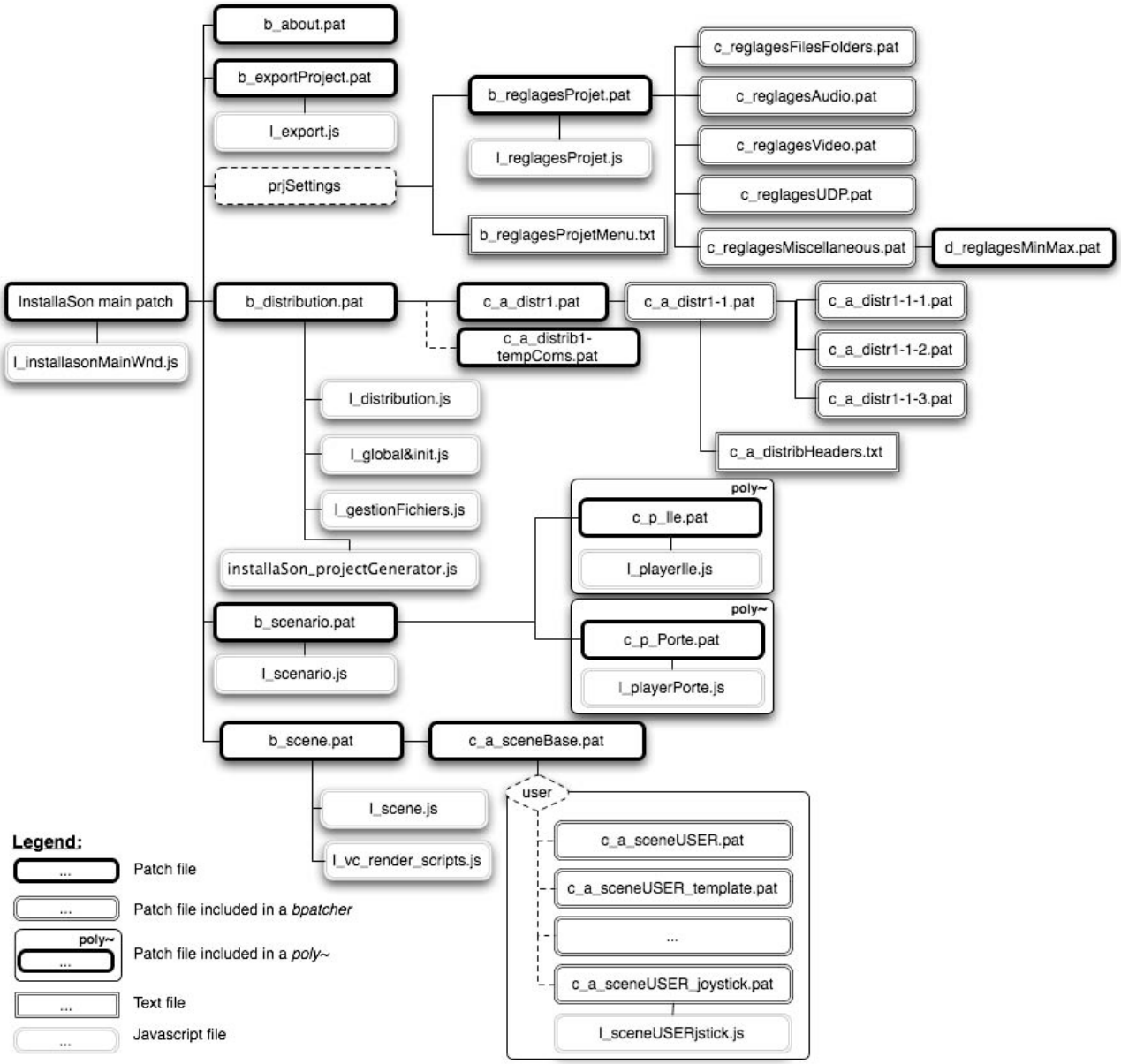

Figure 13. Schéma de liaison entre les différents fichiers qui composent *InstallaSon*.