

TUTORIAL 01

FOR

MIRAGE

Conception : Georges GAGNERÉ, Guy LEVESQUE
Aide programmation : Pedro Soler
© cie incidents memorables 2004



MIRAGE

Objectif : Présentation des différentes étapes pour construire avec le moteur Mirage deux états vidéo simples et les manipuler en temps réel par une interface de contrôle.

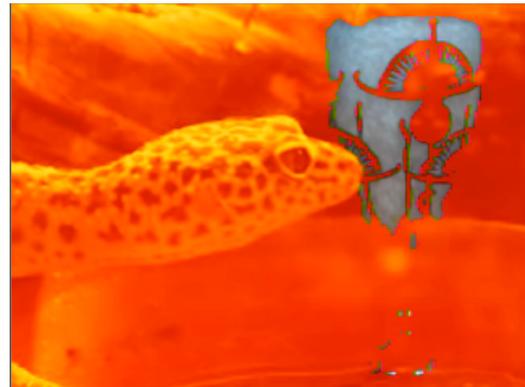
Ce document a été réalisé en collaboration avec Guy Levesque, régisseur vidéo (les tutoriaux détaillés de Mirage sont présents sur www.indidentsmemorables.org/mirage).

Edition

Voici les 2 états à construire (composer) :



BUS A



BUS B

Pour cet exemple, nous allons utiliser 3 médias (2 vidéo, 1 image).



dragonmr.mov



vase.pct



feu.mov

1) Paramétrage des sources

On sélectionne donc trois sources dans le moteur :



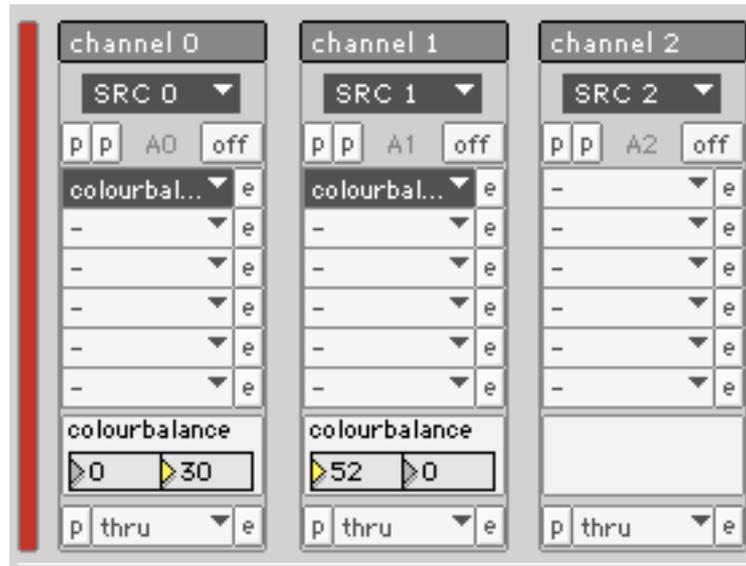
Chaque source présente plusieurs paramètres de réglages. Les séquences vidéo sont des fichiers qui sont sur le disque dur de l'ordinateur sur lequel le moteur est installé (en

l'occurrence feu.mov et dragonmr.mov). De la même manière l'image est un fichier informatique (vase.pct).

2) Paramétrage des bus A et B

Le bus A est une combinaison d'une vidéo (dragonmr.mov) et d'une image (vase.pct) utilisant chacun la moitié de l'image vidéo final. Le dragon est à gauche, le vase à droite. Les Flammes (feu.mov) interagit sur les deux médias.

- BUS A channel (traitement des médias)



On applique à chaque source les effets dont on a besoin : nous utilisons en l'occurrence l'effet colourbalance qui permet de modifier les couleurs d'une image. On peut appliquer aucun effet (comme dans le cas de la source2) ou plusieurs (jusqu'à 7 simultanément).

- BUS A layer (construction de l'état du bus)

A	source	zoom	mask		crop				position		copy	transp														
		zoom	center	x y	mask	blur	left	top	right	bottom	grid	x y	mode	arency												
layer 0	ch_2	P	d	>135.	>81.	>113.	d	off	>0	e	d	>161.	>0.	>320.	>240.	320	240	d	0	>0	>0	max	>0.	d	P	
layer 1	ch_1	P	d	>100.	>220.	>120.	d	off	>0	e	d	>0.	>0.	>159.	>240.	320	240	d	0	>0	>0	max	>0.	d	P	
layer 2	ch_0	P	d	>100.	>160.	>120.	d	off	>0	e	d	>0.	>0.	>320.	>240.	320	240	d	0	>0	>0	blend	>30.	d	P	
layer 3	off	>	p	d	>100.	>160.	>120.	d	off	>0	e	d	>0.	>0.	>320.	>240.	320	240	d	0	>0	>0	blend	>100.	d	P

Exemple de paramétrage d'un layer : nous devons centrer le vase dans la moitié droite, l'agrandir et couper le bord gauche qui ne nous intéresse pas. Nous faisons alors subir en temps réel les transformations suivantes à l'image :

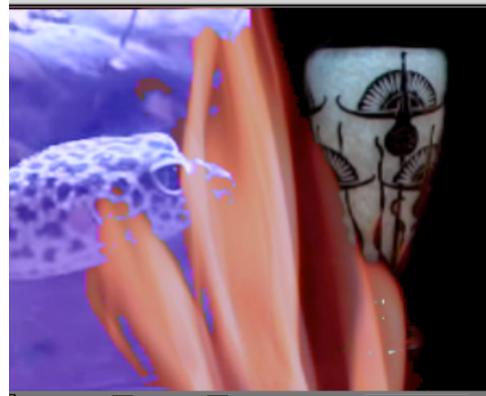
- ZOOM : 135 (agrandissement de l'image d'un facteur 1,35) ;
- CENTER X,Y : 81, 113 (déplacement du centre de l'image de 79 pixels vers la droite et de 7 pixels vers le haut) ;
- CROP : LEFT : 161 (on coupe la gauche de l'image sur une longueur de 161 pixels, pour ne garder qu'une moitié d'image).

Nous réalisons ensuite des transformations analogues sur les deux autres layers pour composer le résultat suivant :

Prévisualisation :



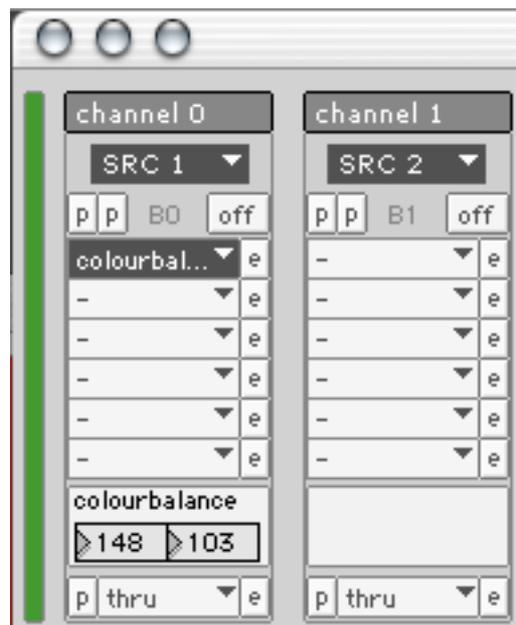
Compositing des layers 1 et 2



Compositing final

Sur le bus B, nous avons les réglages suivants à partir des mêmes sources présélectionnées.

- BUS B (channel)



On n'utilise que deux médias pour la nouvelle composition sur channel 0 et channel 1.

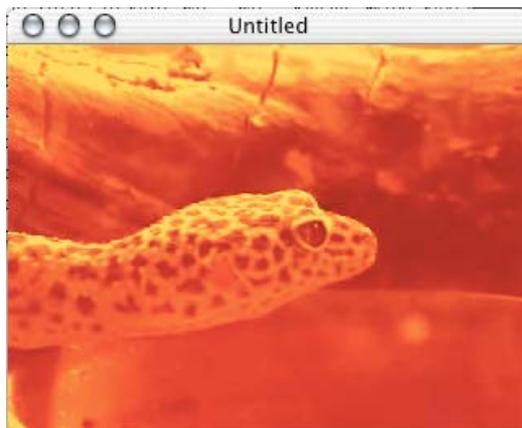
- BUS B (layer)

B	source	zoom	mask	crop	position	copy mode	transp arency
	ch	center x y	mask blur	left top right bottom	grid x y		
layer 0	ch_1 P	d >153. >81. >114.	d off ▾ 0 e	d >161. >0. >320. >240. 320 240	d 0 >0 >0	max	>0. d P
layer 1	ch_0 P	d >100. >160. >120.	d off ▾ 0 e	d >0. >0. >320. >240. 320 240	d 0 >0 >0	blend	>0. d P

On utilise que deux layers.

Et on obtient le résultat suivant :

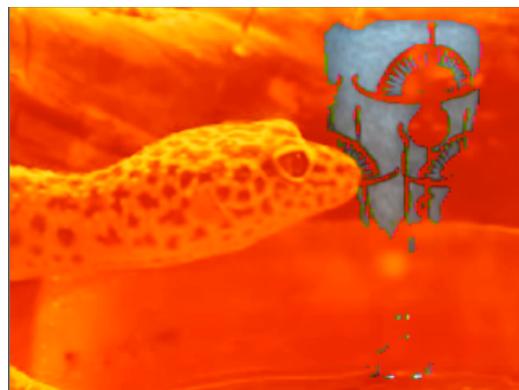
Prévisualisation :



Layer 0



Layer 1



Compositing final

3) SAUVERGARDE DES ETATS (Source, Bus A et B)

Store : sauvegarde toutes les données source comme fichier suivant le numéro choisi

Recall : charge le contenu source du numéro de preset

Read : ouvre un fichier texte sur le disque dur

Write : enregistre un fichier texte sur le disque dur



Vous pouvez vous exercer en utilisant les fichiers de Bus preset et Source preset placés dans le dossier /tutorial_main.crossfade : BusA preset 1, BusBpreset 2, Src preset 5. (read puis recall 1 pour l'état du bus A de ce tutorial, de même pour les autres preset)

Contrôle

A partir des deux états vidéo que nous avons édité avec le moteur, nous pouvons désormais envisager une phase de manipulation. Nous allons mettre en œuvre un langage de programmation pour envoyer des instructions à Mirage et transformer les états vidéo précédemment construit. Il faut imaginer que tout ce que nous avons construit est contrôlable en temps réel.

Mirage a permis d'organiser les matériaux selon trois étapes : choix des sources, choix des effets, choix du compositing.

Nous allons successivement décrire cinq manipulations :

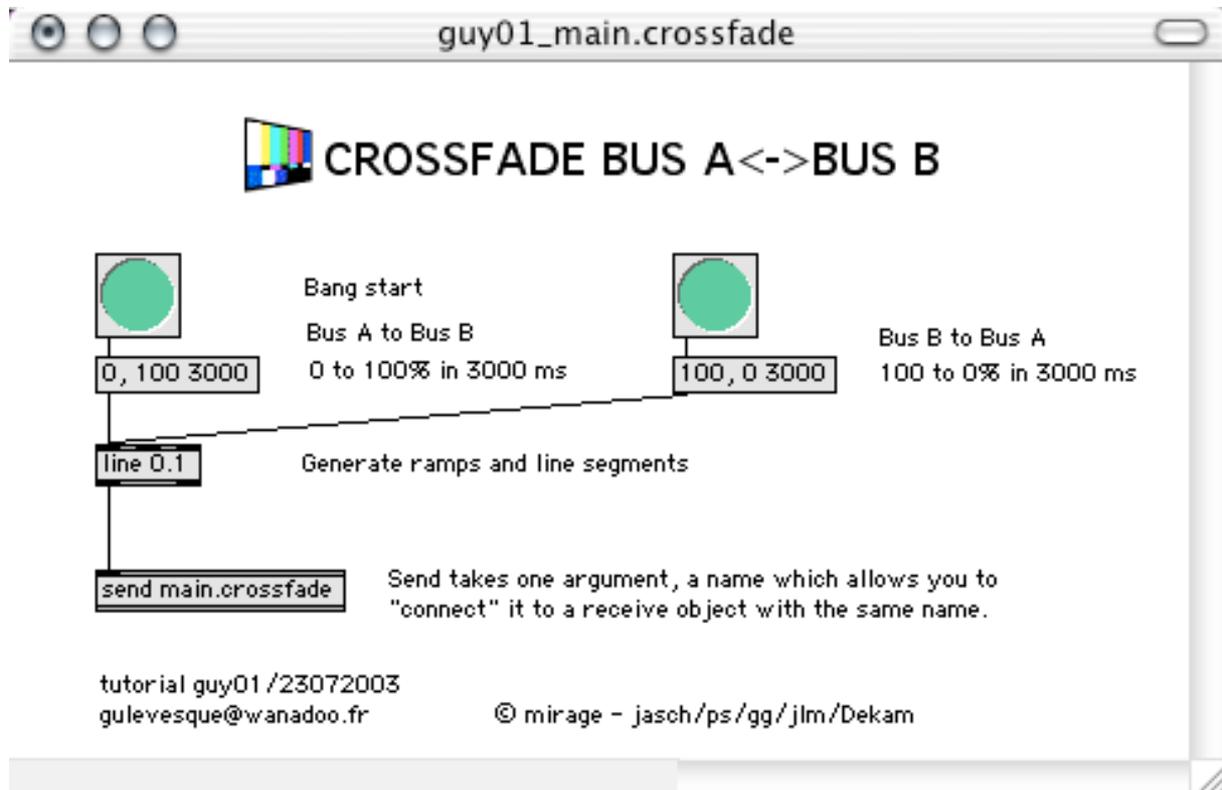
- passage d'un bus à l'autre
- changement d'une photo dans les sources
- changement d'un effet (flip vertical du vase)
- changement du compositing sur les channels : effet ROTATE (sur le feu)
- changement du compositing sur le déplacement des layers (sur le dragon)
- Agencement des manipulations successives pour la conduite (scénario)

PASSAGE D'UN BUS À L'AUTRE

Création de patch MAX : main.crossfade

L'idée est de passer du Bus A au Bus B en un temps donné, paramétrable par nous.

L'adresse sous M.I.R.A.G.E est : `main.crossfade <float>` ; 0-100%, 0% bus A, 100% bus B



BANG : permet d'envoyer le signal manuellement pour le départ de l'effet

0, 100 3000 : de 0 à 100 en 3000ms

line : permet de générer la progression de la valeur départ à la valeur finale.

send main.crossfade : envoi à l'adresse main.crossfade

CHANGEMENT D'UNE PHOTO DANS LES SOURCES

Création du patch MAX : Still.swap

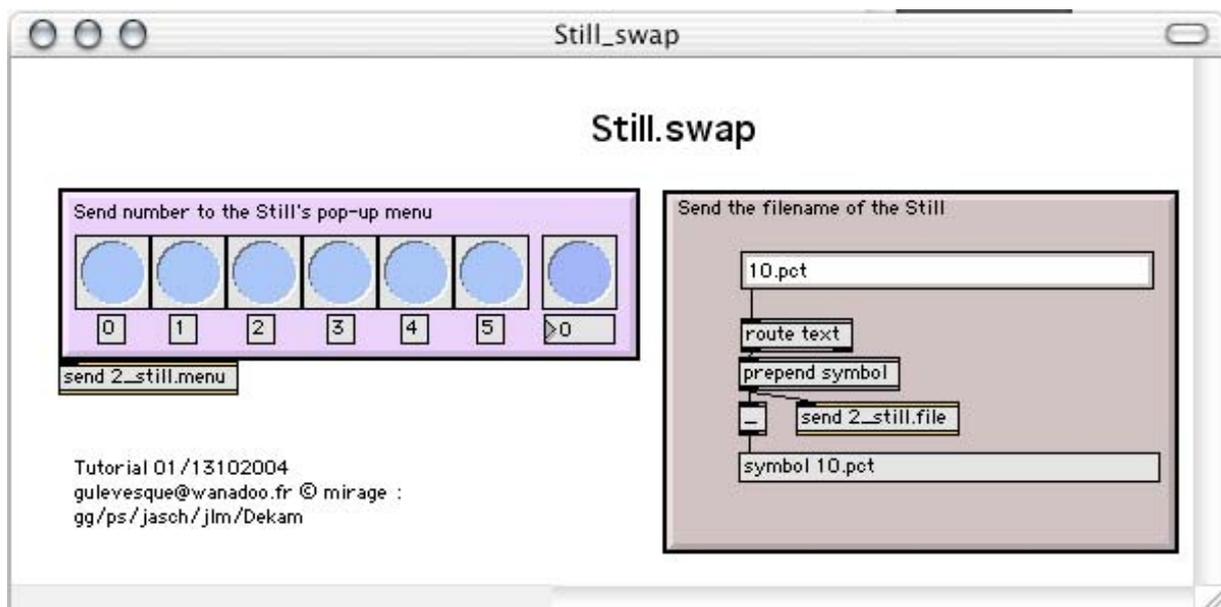
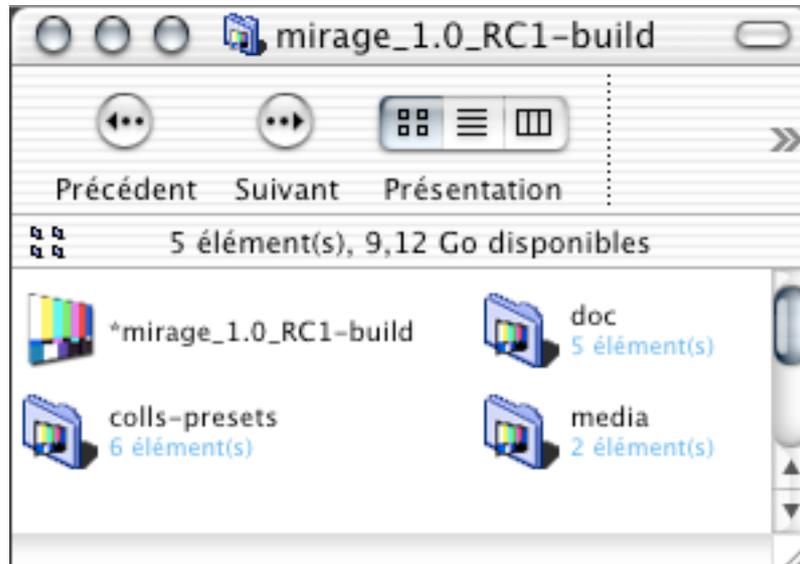
Les adresses sous M.I.R.A.G.E sont :

`N_still.file <symbol>` **pict file to load into stillstream** ou `N_still.menu <int>` **access to stillstream menu**

N étant le numéro de module de la source, en l'occurrence pour notre exemple, le Still est affectée à la source 2.

2 possibilités s'offrent à nous pour changer un fichier image dynamiquement : soit l'appeler directement par son nom de fichier avec l'adresse `N_still.file <symbol>`, soit appeler par son numéro dans l'ordre du menu déroulant de la source Still avec l'adresse `N_still.menu <int>`.

Remarque : les fichiers media (movie ou still) doivent être placés dans le dossier /media de l'application mirage avant que mirage ne soit ouvert (si vous rajoutez des stills ou movies, il faut quitter mirage et le relancer pour voir apparaître les noms des fichiers rajoutés).



CHANGEMENT D'UN EFFET : FLIP VERTICAL DU VASE SUR LE BUS A Création du patch : Flip.vase

L'adresse M.I.R.A.G.E est :

A0_flip.h <int> (0/1), **horizontal flip on Bus A/Layer 0**

A0_flip.v <int> (0/1), **vertical flip on Bus A/Layer 0**

Affectation de l'effet FLIP sur le vase : dans mirage, choisir dans le menu déroulant des effets le FLIP :



channel 2



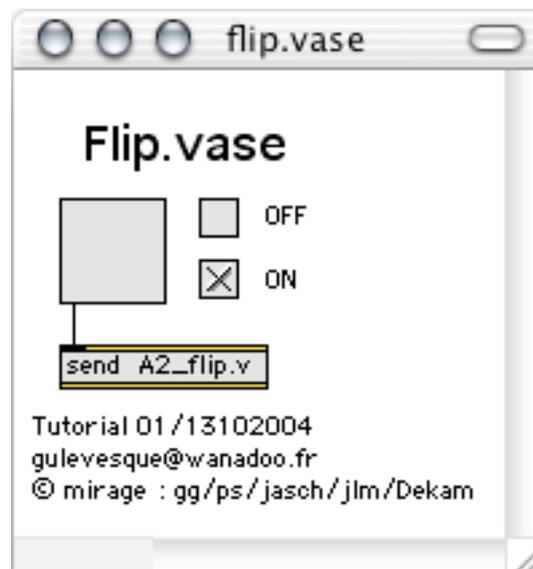
Avant



Après

Pour notre tutorial, nous allons appliquer l'effet FLIP sur le vase qui se trouve sur la Channel 2 du bus A. L'adresse devient A2_flip.v <int>.

Le patch se présente sous la forme :



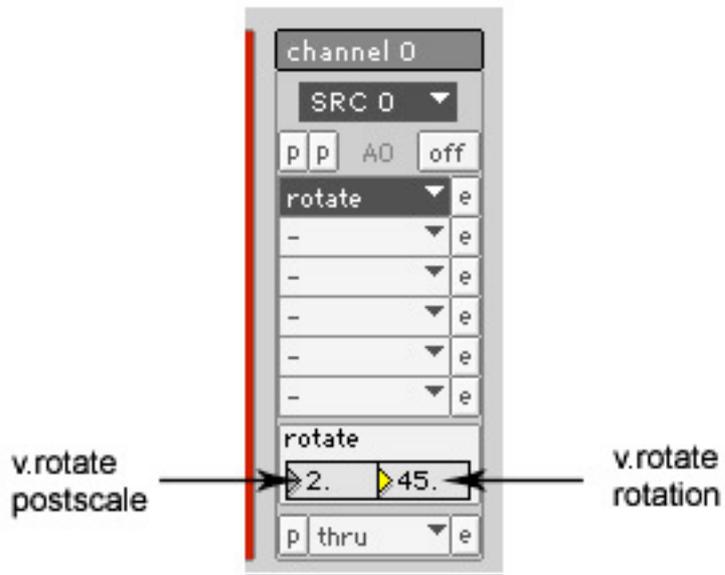
CHANGEMENT DU COMPOSITING AVEC LES CHANNELS (effet ROTATE)

Création d'un patch : channel.rotate

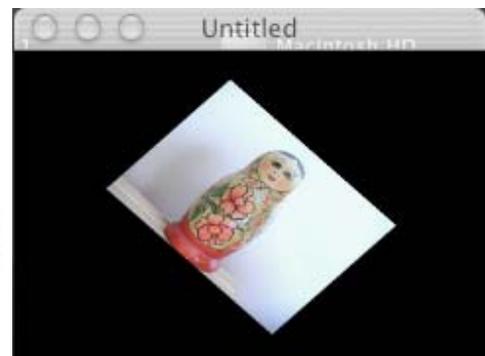
Les adresses M.I.R.A.G.E sont :

A0_rotate.postscale <float> **post scale (1.0 = normal)**

A0_rotate.angle <float> **rotation (degrés)**



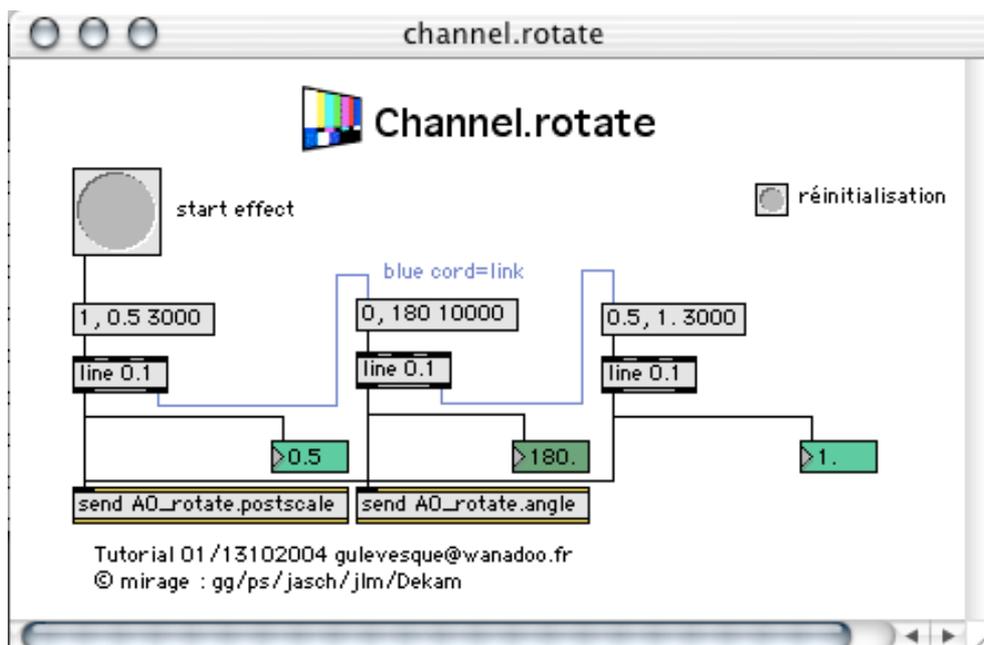
mirage effet ROTATE



résultat du ROTATE

Pour ce tutorial, nous allons appliquer l'effet ROTATE sur le feu. Le résultat que nous souhaitons est une rotation de 180° des flammes. Cette rotation doit se faire en 10 secondes.

Le patch sous MAX se compose des objets suivants :

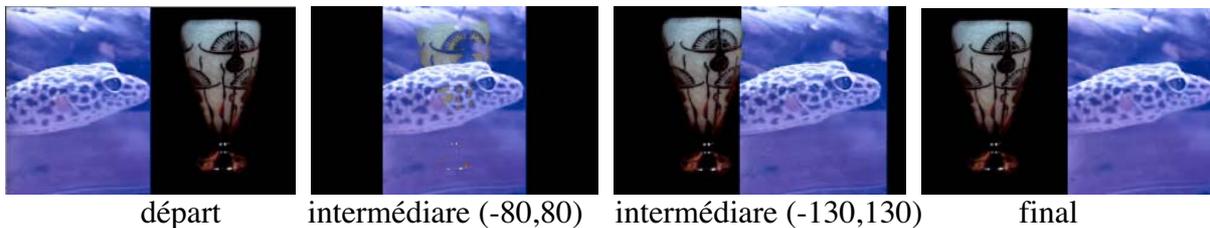
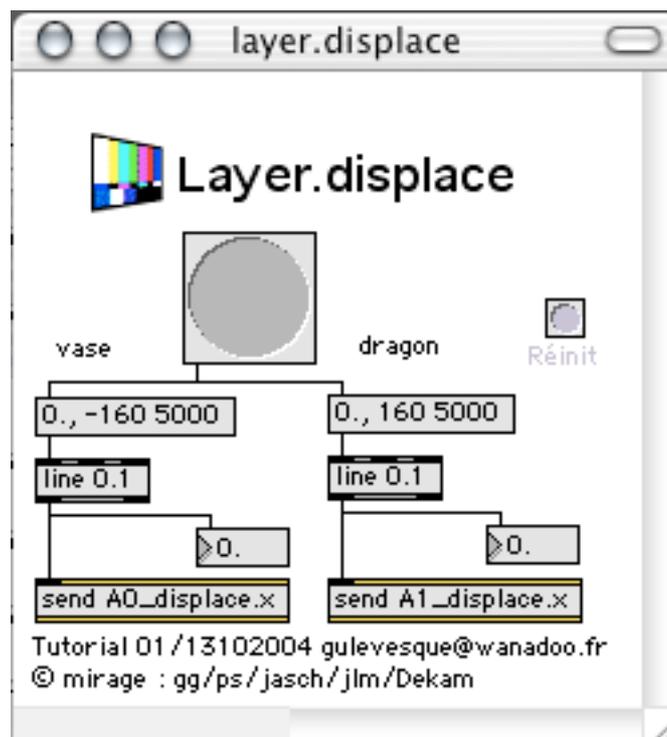


CHANGEMENT DU COMPOSITING AVEC LES LAYERS (commande displace) Création d'un patch : layer.displace

Permutation des medias Dragon & Vase en 5 secondes. Le dragon va de gauche à droite pendant que le vase va de droite à gauche.

Adresse sous M.I.R.A.G.E :

A0_displace.x <int> : **displace x value (pixel)**



AGENCEMENT DES MANIPULATIONS SUCCESSIVES : SCÉNARIO & CONDUITE Création d'un patch : tutorial01.conduite

Introduction à la CONDUITE

- envoi des effets aux tops
- visualisation rapide de la succession des effets

Cette conduite reprend les presets source et bus du tutorial GUY01 téléchargeable à l'adresse suivante : <http://perso.wanadoo.fr/musica.online/mirage>

Scénario :

Le public rentre dans l'espace scénique, il voit en place sur l'écran de projection le compositing du bus A (dragon/vase/flammes). Aucun comédien n'est sur la scène.

A l'entrée du comédien (TOP 1), on passe du bus A vers le bus B, c'est le premier TOP pour la conduite. Arrivée au centre du plateau de théâtre le comédien joue le texte.

Le dragon et le vase

Par un beau matin de printemps, un dragon se reveilla de sa cachette, après avoir passé l'hiver dans sa confortable maison. Au pied de sa porte (TOP 2 : passage du bus B vers le bus A), il y découvrit un vase ! Étonné, le dragon inspecta chaque recoin de ce mystérieux vase, aucune inscription, « qui a perdu ce magnifique vase ? » se demanda-t-il.

Toujours intrigué, il mit son œil dans le trou du vase pour voir ce qu'il y avait à l'intérieur et surprise (TOP 3 : Flip vertical du vase), une magnifique fleur s'y trouva !

Il prit le vase à l'envers (TOP 4 : changement de l'image du vase en fleur), et secoua le vase afin de la faire sortir. Après quelques essais, il y parvint.

Content de voir cette fleur encore pleine de vitalité, il poussa un grognement d'extase et souffla un bon coup de feu par sa bouche. (TOP 5 : rotation des flammes)

Le dragon prit la fleur et la mit (TOP 6 : Permutation du dragon et de la fleur) dans de l'eau à côté de son lit en espérant la voir grandir à ses côtés. Il partit à travers la forêt chercher un peu de nourriture pour lui et la fleur.

Identification des points de conduite (TOP) :

- 1- Passage du bus A vers le bus B
- 2- Retour du bus B vers le bus A
- 3- Flip vertical du vase
- 4- Changement de l'image du vase en fleur
- 5- Le feu fait une rotation
- 6- Permutation du dragon et de la fleur

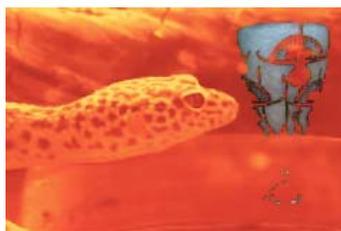
Écriture du patch de conduite :

Conduite à construire, étape par étape :

Nous avons créé des patchs correspondant aux différentes tâches à effectuer. Voici les étapes par Top du petit spectacle.



Entrée public



TOP 1



TOP 2



TOP 3



TOP 4



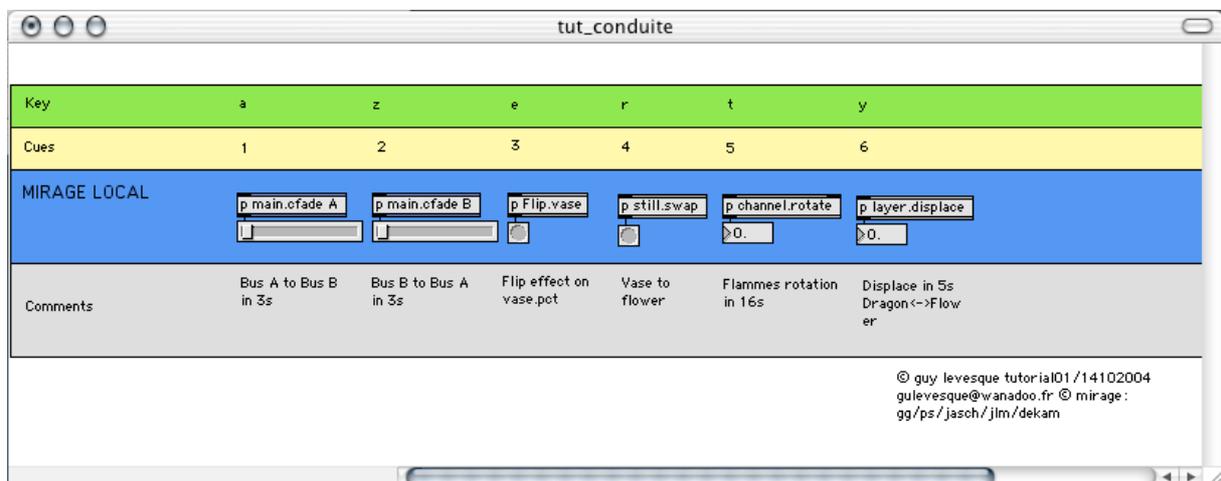
TOP 5



TOP 6



Final



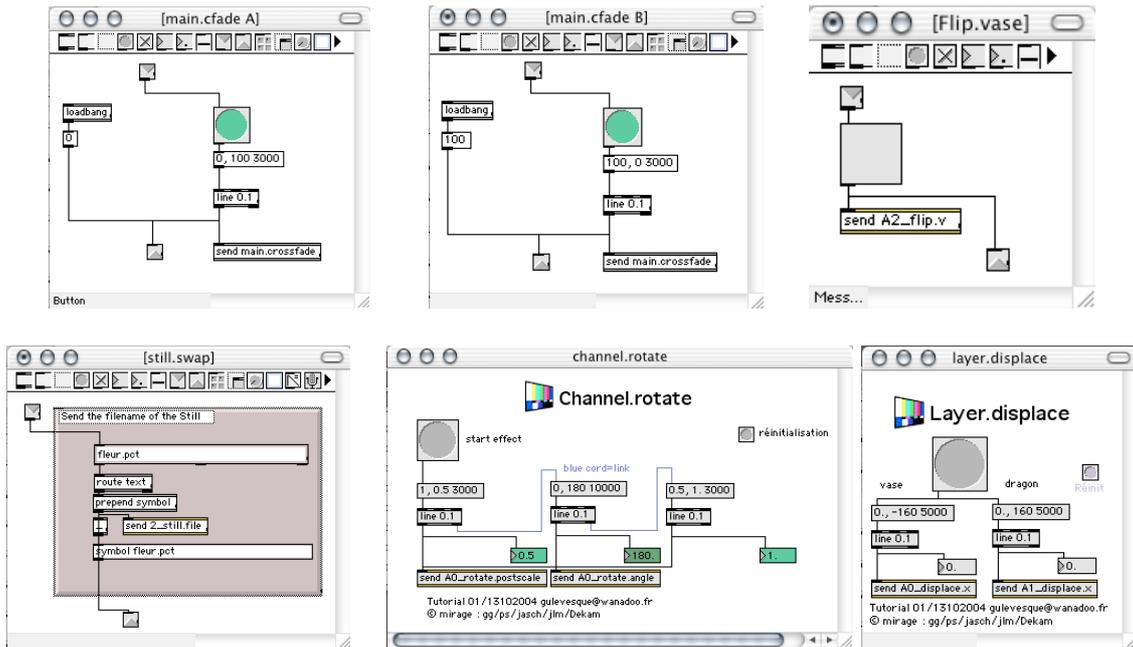
Patch de conduite

Le patch de conduite se présente sous la forme de tâche successive à effectuer par le régisseur. Des fonctions peuvent être programmés afin de rendre la conduite jouable par le régisseur.

Ici, on utilise une fonction qui assigne aux lettres du clavier de l'ordinateur chaque top. Le régisseur peut alors déclencher ses tops avec les lettres a,z,e,r,t,y. On peut paramétrer le clavier suivant nos besoins ou utiliser un clavier maître MIDI pour déclencher les tops.

Chaque top correspond a une boite (patcher) contenant les données à envoyer au moteur mirage.

Les images suivantes illustrent les patchers utilisés pour la conduite. Ils proviennent tous des différentes étapes abordées depuis le début de ce document.



patchers

Conclusion

L'outil mirage utilisé comme application vidéo permet le compositing d'image. La mise en place d'une interface de contrôle, dialoguant avec le moteur mirage, ouvre les portes du contrôle dynamique de la vidéo et de son interaction avec le plateau théâtral.

Ce nouvel outil (moteur/interface) demande une méthode de construction des environnements vidéo et de la souplesse afin de permettre de faire rencontrer le jeu des comédiens et la scénographie/mise en scène/technique.

Nouvel outil/Nouvelle méthode de travail (compétence)

Le moteur rassemble des opérations complexes et des modes de combinaisons de ces opérations que l'on peut contrôler de l'« extérieur » par un système d'adresses. On définit ainsi ce qu'on appelle un protocole d'adressage, que l'utilisateur agence à l'aide d'un langage de programmation élémentaire sur l'interface de contrôle. Le moteur est donc un logiciel dont la programmation complexe est réalisée par un développeur confirmé, et qui offre des possibilités de contrôle simplifié pour des utilisateurs extérieurs. Ces utilisateurs n'ont pas besoin de savoir comment fonctionne le moteur, ils l'utilisent.

L'utilisation requiert cependant la maîtrise d'un langage d'assemblage d'adresses, qui peut être un langage objet, simple d'utilisation, permettant de visualiser graphiquement l'agencement des instructions.

Ainsi, à partir de l'architecture Moteur/interface de contrôle, un double dialogue peut s'enclencher entre trois niveaux de compétence numérique : celle du régisseur-utilisateur, celle de l'artiste-programmeur et celle du développeur.

Au niveau de l'interface de contrôle, le régisseur et le programmeur doivent se répartir la manipulation et la fabrication des matériaux à partir des fonctionnalités existantes du moteur. Ils doivent décider ensemble des prérogatives de chacun dans la mise en place d'une conduite d'effets générant des images.

Si le régisseur acquiert une bonne compétence numérique, la réflexion peut s'ouvrir sur les conditions pour améliorer le déroulement des répétitions et inventer des méthodes d'élaboration souple et rapide de dispositifs pouvant répondre aux sollicitations des acteurs et du metteur en scène. La réflexion sur le temps réel du temps réel peut alors commencer de se concrétiser.

Au niveau du moteur, le régisseur et le programmeur peuvent réfléchir avec le développeur du moteur aux nouvelles fonctionnalités à intégrer pour améliorer l'ergonomie du contrôle par l'interface.