

## **PIXELMAPS manual**

### **INDICE**

<b>QUE ES PIXELMAPS .....</b>	<b>2</b>
<b>JERARQUIA EN PIXELMAPS .....</b>	<b>3</b>
<b>CONTENIDOS. ESTRUCTURA JERARQUICA PIXELMAPS .....</b>	<b>4</b>
<b>DOCUMENTO XML. ESTRUCTURA JERARQUICA PIXELMAPS.....</b>	<b>6</b>
<b>TEMPLATES. INTRODUCCIÓN .....</b>	<b>11</b>
<b>TABLA HEXADECIMAL PARA CADA PIXEL DE UNA TEMPLATE EN PIXMAP .....</b>	<b>12</b>
<b>TEMPLATES MOSAICO .....</b>	<b>14</b>
<b>CREAR LAS TEMPLATES con el programa INKSCAPE.....</b>	<b>15</b>
<b>TABLA DE MENSAJES OSC EN PIXELMAPS.....</b>	<b>21</b>
<b>ANEXO A PIXMAPS Y TEMPLATES .....</b>	<b>24</b>

## QUE ES PIXELMAPS

PixelMaps es una librería basada en Open Frameworks, en adelante oF, [<http://openframeworks.cc/>]. En 2010 Javier Chavarri introdujo el uso de algunas librerías de oF en la performance Cotrone [<http://marceliantunez.com/work/cotrone/>]. En ella se utilizaba oF como interfaz para detectar rostros, en la escena Spizzi o para reconocer las imágenes de unas tarjetas postales en la escena Conde. Al mostrarlas en cámara las postales activaban unas animaciones en alfa que emergían sobre la imagen que daba la postal en pantalla. Más tarde en 2011 Javier hizo las primeras pruebas de integrar mosaicos de vídeos en oF en el workshop MITOTICA realizado en Costa Rica. Sergi Lario retomó el proyecto oF en 2012, que propuso llamar a esta librería PIXMAPS y quien ha desarrollado hasta la fecha diferentes versiones del programa que sistematizan la gestión de los vídeos, como veremos. La versión que presentamos aquí es solo una parte de los diferentes ejecutables de PIXMAPS que hemos desarrollado específicamente para distintos proyectos. Nuestra idea es, en un futuro próximo, añadir los elementos que ya han sido desarrollados en un solo ejecutable.

La versión de PIXMAPS febrero 2017 puede aplicar varios efectos de posición, mirror, color y sombreado a un vídeo o a un grupo de vídeos, gracias a la numeración hexadecimal de un pixel RGBA, Red Green Blue Alpha. Con estos grupos de vídeos se pueden confeccionar mosaicos de 1 solo pixel hasta un máximo de 32x 24 vídeos. Con la técnica del pixel hexadecimal se puede elegir de un banco determinado de vídeos aquellos que deseemos para componer un mosaico. No obstante la complejidad del mosaico, es decir más vídeos en una sola imagen, está en relación a la definición y la longitud temporal de los vídeos que contiene, a más cantidad píxeles en el mosaico menos definición y/o tiempo de los vídeos. Podemos también sobreponer una o alternativamente varias cámaras IP sobre los vídeos que estamos reproduciendo y reproducir animaciones en alfa sobre la cámara o sobre los vídeos. En breve incorporaremos el sistema de captura de vídeos en tiempo real lo que nos permitirá trabajar de forma dinámica con capturas de vídeo hechas al instante. PIXMAPS tiene prefijadas una serie de etiquetas en Open Sound Control, en adelante OSC, [<http://opensoundcontrol.org/>] que permiten que este programa se ejecute desde POL, pero también desde otros programas que hablen OSC como por ejemplo Pure Data, en adelante PD, [<https://puredata.info/>] o con Max/SP.

PIXMAPS es el programa que proponemos en Systorgy para la gestión del vídeo y animaciones alfa pero existen otras alternativas como por ejemplo Flash. Añadimos un ejemplo del uso de Flash en una de las scenes de nuestro ejemplo la performance POL\_MATRIX.pol. En ella veremos la forma para el control secuencial de vídeo en formato \*.flv de Flash.

## JERARQUIA EN PIXELMAPS

Utilizando una metáfora, el lugar donde disponemos los datos (vídeos, píxeles hexadecimales, animaciones, capturas) es el almacén de archivos y el documento escrito en XML es el archivo que los gestiona. El archivero ordena por su nombre, su ruta en el almacén y en ocasiones por un número, a los distintos archivos y este orden es el que utiliza POL para gestionar interactivamente estos archivos. Si se altera alguna de los elementos que componen este orden – nombre, ruta, número –, se ejecutará otro contenido o simplemente no funcionará. Como verás algunas de las tareas de PIXELMAPS, como por ejemplo la confección de los píxeles hexadecimales, no son inmediatas y hay que comprobarlas con todo el sistema montado o con un simulador, para cerciorarnos de su eficacia. La figura del archivero en PIXMAPS es un documento de texto que llamamos **appSettings.xml**. Para modificarlo y adaptarlo a nuestras necesidades aconsejamos utilizar la aplicación Notepad ++. Puedes descargarte libremente Notepad ++ aquí: [<https://notepad-plus-plus.org/>].

Así pues los bancos de vídeos, animaciones, cámaras y píxeles hexadecimales y el sistema para controlarlos, el archivo **appSettings.xml**, mantienen una relación jerárquica absolutamente dependiente. Cualquier cambio en ambos puede alterar o incluso impedir la reproducción de los vídeos, hay que estar muy atento a este orden.

Llamaremos en **appSettings.xml** al pixel hexadecimal, 1x1, o sus mosaicos, de 1x2 hasta 32x24, **Templates [tmp]**, a su organización en **grupos [groups]**, a los **vídeos [video]** y a los conjuntos de **videos [bank]**. A las **animaciones**, basadas en secuencias de png, **[alfa]** y a los conjuntos de **capturas** directorios **[dir]**. Aunque los **dir** están operativos para su reproducción, la forma de capturarlos por el momento no. Llamaremos las **cámaras** por su nombre en nuestro ejemplo **[showMicCam]**.

Para facilitar la comprensión de este manual y de los demás elementos que componen el proyecto Systorgy, nos basamos en el entorno POL\_Matrix.pol con todos los elementos necesarios, incluidos vídeos de ejemplo, para ejecutar PIXMAPS. ATENCIÓN la versión de POL\_MATRIX.pol final no coincide con algunos de los elementos que describimos en este manual, considera las nuevas carpetas como equivalentes al ejemplo que comentamos a continuación. Encontraras en el zip POL\_Matrix, en el site [<https://systorgy.hangar.org>]. Utilizaremos este ejemplo como performance para describir en adelante PIXELMAPS.

## CONTENIDOS. ESTRUCTURA JERARQUICA PIXELMAPS

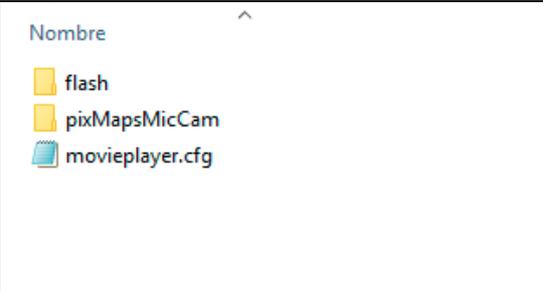
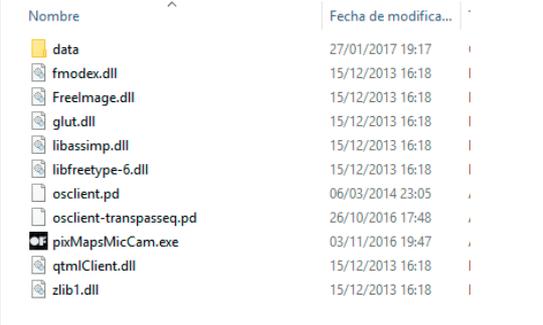
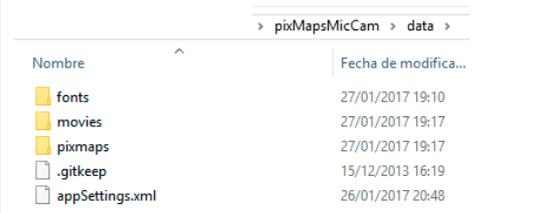
Como ya hemos dicho tenemos dos ámbitos el “almacén”, donde están los contenidos y que como veremos inmediatamente, están sujetos a una ruta determinada, y que el “archivero” es el documento XML **appSettings.xml** y que nos permite configurar los elementos de esta ruta para utilizarlos en nuestra performance.

Esta es la forma de organizar la ruta de los archivos en el “almacén” PIXELMAPS. Recuerda que tenemos esta ruta y su carpeta principal, aquí **systorgy\_pixmaps**, en el ordenador que reproduzca los gráficos y que en nuestro ejemplo es el laptop destinado a ello.

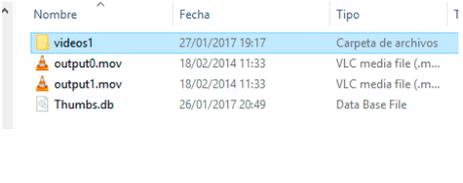
El nombre de esta carpeta lo puedes cambiar pero ten en cuenta que tendrás que escribir en el panel de configuración, **devices configuration panel**, de POL Setup en el apartado **PDM config**. la ruta completa hasta la carpeta . Ten en cuenta que el PDM, Pol Display Manager se ejecuta en el ordenador de gráficos y que la ruta arranca desde esta máquina. Puedes ver más información en manual POL Setup, en el apartado: NUMBER OF POL DISPLAY MANAGERS / PDM CONFIG.

Así en la ruta dada en nuestro ejemplo es **D:\grafic\_performances\systorgy\_pixmaps\** que en adelante describimos en forma abreviada como **D:\...\**

Los archivos se organizan así:

imagen	Ruta y descripción
	<p><b>D:\...\ systorgy_pixmaps\</b></p> <p>Al abrir la carpeta general de la performance de nuestro ejemplo encontrarás 2 carpetas raíz: la de <b>flash</b> relacionado con los contenidos que ejecuta este programa y la <b>pixMapsMicCam</b> relacionada con los contenidos que describimos en este manual.</p> <p>Aquí es donde aconsejamos situar las carpetas relacionadas con otros programas de gráficos en el caso que los uses</p>
	<p><b>D:\...\ systorgy_pixmaps\ pixMapsMicCam</b></p> <p>En esta carpeta están las DLL del ejecutable de programa <b>pixMapsMicCam.exe</b> y la carpeta <b>data</b>.</p>
	<p><b>D:\...\ pixMapsMicCam\data</b></p> <p>Aquí están situadas las carpetas de fonts, que no describimos, movies y pixmaps, estas dos últimas las vemos en adelante. También está el “archivero” <b>appSettings.xml</b>. Aquí es donde tienes que venir a buscarlo para abrirlo y configurarlo como se describe en adelante.</p>

	<p><b>D:\...\ pixMapsMicCam\data\pixmaps</b></p> <p>En el interior de la carpeta <b>pixmaps</b> están otras carpetas, son los <b>groups</b> que contienen las <b>templates</b> . Algunos de estas carpetas o groups os pueden servir de patrón modelo para nuevas performances. Veremos esto en adelante. Como verás las <b>templates</b> son el marco donde se reproduce el vídeo, a excepción de la cámara y de las animaciones, todos los vídeos necesitan al menos de una template para poder reproducirse.</p>
	<p><b>D:\...\ pixMapsMicCam\data\movies</b></p> <p>Hemos cerrado la carpeta anterior pixmaps y abrimos la de <b>movies</b>. Aquí descubrimos la jerarquía de carpetas: alfas, Banks y captures. En estas carpetas almacenaremos los contenidos de las animaciones, los vídeos y las capturas</p>
	<p><b>D:\...\ pixMapsMicCam\data\movies\alfas</b></p> <p>La estructura muestra 4 ejemplos de carpetas con contenidos de png en alfa que PIXMAPS abre en RAM y ejecuta. Los nombres de cada una de estas carpetas esta dinámico. En este ejemplo nosotros lo hemos determinado en función de la narrativa y sus contenidos.</p>
	<p><b>D:\...\data\movies\alfas\02_arribo_saluda</b></p> <p>Aquí abrimos, a modo de ejemplo, la carpeta <b>02_arribo_saluda</b>, esta carpeta, de la que vemos sólo algunos png's, contiene 41 png's que permiten crear una animación en loop de aprox 2 segundos. Con esta técnica y puesto que necesitas cargar los png en RAM, el número de png por carpeta está en función de la capacidad de proceso del ordenador de gráficos.</p>
	<p><b>D:\...\ pixMapsMicCam\data\movies\banks</b></p> <p>La carpeta banks contiene un número determinado de carpetas que llamamos bank y que a su vez contienen vídeos. Cada bank contiene uno o varios vídeos. Si la template que reproduce los vídeos es un mosaico de pixeles y estos están configurados para llamar a varios vídeos, pondremos todos esos vídeos dentro de un solo bank.</p>
	<p><b>D:\...\ data\movies\banks\MURAL43</b></p> <p>Aquí hemos abierto el bank MURAL43 y vemos el vídeo Mural43.mov. Este es un ejemplo de bank con un solo vídeo.</p>

	<p><b>D:\..\movies\banks\ ANGURIAESPAGUETTILENGACARN</b></p> <p>Hemos abierto ANGURIAESPAGUETTILENGACARN observa que aquí hay 8 vídeos. Podemos por tanto llamar a cada uno de estos vídeos por su número, según el orden del primero al último [1,2,3...8]. Podemos llamarlos desde POL y como veremos desde las templates.</p>
	<p><b>D:\...\pixMapsMicCam\data\movies\captures</b></p> <p>En la carpeta captures se sitúan las subcarpetas relacionadas con las capturas y que conocemos como <b>dir</b>. Cuando la captura es efectiva la cámara graba archivos que llama <b>output0.mov</b> y consiguientes hasta <b>output15.mov</b>, en carpetas que se llaman videos1 y consiguientes.</p>

## DOCUMENTO XML. ESTRUCTURA JERARQUICA PIXELMAPS

Como comentamos el “archivero” es el documento **appSettings.xml** que abriremos con la aplicación Notepad ++. De las capturas de pantalla de este programa provienen las imágenes que ilustran los siguientes ejemplos. Utiliza este programa para editar XML, descargarlo libremente en [<https://notepad-plus-plus.org/>].

El archivo **appSettings.xml** se encuentra dentro de la carpeta data que a su vez está en la carpeta pixMapsMicCam. En nuestro ejemplo:

**D:\grafic\_performances\systorgy\_pixmaps\ pixMapsMicCam\data**

XML es el acrónimo de Extensible Markup Language puedes obtener más información de este lenguaje en [[https://es.wikipedia.org/wiki/Extensible\\_Markup\\_Language](https://es.wikipedia.org/wiki/Extensible_Markup_Language)]

El programa es un archivo de texto en el que vamos a replicar la jerarquía de contenidos de nuestra performance por tanto puede contener cientos de líneas. Hay que ser muy cuidadoso en la edición de estas líneas pues cualquier error, un espacio, un nombre incompleto, etc.... no sólo no permiten reproducir el archivo donde se encuentre el error sino probablemente los siguientes elementos del grupo.

En realidad aquí configuramos las cámaras, las etiquetas de los distintos directorios PIXGROUP, VIDEOBANK, ALFAVIDEOBANKS y **VIDEODIR**, con sus contenidos respectivamente pixmaps, videos, secuencias png’s y capturas de vídeo. También configuramos los puertos y dirección IP para permitir la comunicación OSC con otras aplicaciones.

El programa es bastante intuitivo y no vamos a detallar aquí como funciona salvo algunas reglas básicas útiles para modificar el archivo **appSettings.xml** y adaptarlo a nuestras necesidades.

### Comentar renglones para dejarlos inoperativos.

Siempre que añadamos `<!--` al principio de un renglón, o de un escrito y al final esto `-->` deja de tener funcionalidad en el sistema. Método que utilizamos también para añadir una nota.

Observa este ejemplo en el que se ha añadido esta nota:

```
<!-- ip for the computer waiting to receive data from the application -->
```

### Como distinguir un conjunto de elementos.

Nos permite ver también el inicio y el final de un conjunto jerárquico, – group, bank, alfa, dir – así si damos doble clic a la cabecera por ejemplo de un conjunto de templates, que reconoceremos por la palabra PIXGROUP, y que en programa aparece como `<PIXGROUP>` se ennegrecerá simultáneamente a su final que es `</PIXGROUP>`. Observa que se diferencia entre la cabecera y el final es un slash /.

### Como minimizar un conjunto de la jerarquía.

Verás que junto a la cabecera de un conjunto, `<PIXGROUP>` por ejemplo, hay un cuadradillo con un guión [–] si lo clicas se cerrará el conjunto hasta el final `</PIXGROUP>` y el cuadradillo aparece con un signo más [+]. Clicando sobre el + restituyes la visibilidad del conjunto.

Veamos algunos ejemplos en `appSettings.xml` :

#### INICIO

Este es el inicio del documento y contiene el elemento principal que es `<CONFIG>` que se cierra al final del documento con `</CONFIG>`. El grupo `<OSC>` que contiene la IP y puertos para enviar y recibir mensajes OSC. EL grupo `<GUNCAM_URL>` indica la IP de la cámara que tenemos en el sistema. El grupo `<MICCAM>` indica la posición (x, y) y las dimensiones donde mostrar la cámara. En otras versiones de este programa podemos elegir más de una por performance.

## PIXMAPS y PIXGROUPS

```

15
16
17 <!-- list of grouped pixmaps templates: paths from data/pixmaps -->
18 <PIXMAPS>
19
20 <PIXGROUP> <!--01-->
21 <PIXMAP>1x1v01/f0.png</PIXMAP>
22 <PIXMAP>1x1v01/f1.png</PIXMAP>
23 <PIXMAP>1x1v01/f2.png</PIXMAP>
24 <PIXMAP>1x1v01/f3.png</PIXMAP>
25 <PIXMAP>1x1v01/f4.png</PIXMAP>
26 <PIXMAP>1x1v01/f5.png</PIXMAP>
27 <PIXMAP>1x1v01/f6.png</PIXMAP>
28 <PIXMAP>1x1v01/f7.png</PIXMAP>
29 <PIXMAP>1x1v01/f8.png</PIXMAP>
30 <PIXMAP>1x1v01/f9.png</PIXMAP>
31 <PIXMAP>1x1v01/fa.png</PIXMAP>
32 <PIXMAP>1x1v01/fb.png</PIXMAP>
33 <PIXMAP>1x1v01/fc.png</PIXMAP>
34 <PIXMAP>1x1v01/fd.png</PIXMAP>
35 <PIXMAP>1x1v01/fe.png</PIXMAP>
36 <PIXMAP>1x1v01/ff.png</PIXMAP>
37 </PIXGROUP>
<!-- 1x1 -->

```

## PIXMAPS, PIXEGROUPS y TEMPLATES

En el primer renglón de esta sección vemos la nota:

```
<!-- list of grouped pixmaps templates: paths from data/pixmaps -->
```

Y que nos indica la lista de grupos de pixmaps.

A continuación vemos <PIXMAPS> que contienen todo el conjunto de pixgroups y que se cierra con </PIXMAPS>

Tres renglones más abajo vemos un pixgroup, <PIXGROUP> <!--01-->, que contiene 16 templates. El comentario <!--01--> nos indica la posición del group en la jerarquía del conjunto de pixgroups, y en este caso es el 1. Este comentario solo nos sirve para visualizar en la lista el número de pixgroup que también podríamos hallar contando. Así el primer pixgroup es valor 1, el segundo el 2 etc. Utilizaremos esta posición para llamarlos en POL como veremos en la performance POL\_MATRIX.pol

El pixgroup 1 contiene templates de 1x1 es decir de un solo pixel y por tanto de un solo vídeo en todos sus efectos. Observa que el renglón de este pixgroup

<PIXMAP>1x1v01/f0.png</PIXMAP>, contiene el nombre de la carpeta donde está la template y que es 1x1v01, y separado por un slash /, el nombre de la template es f0.png.

Este pixgroup que tal como se muestra tiene su equivalente dentro de la carpeta del archivo D:\...\pixMapsMicCam\data\xixmaps, contiene las 16 templates que contienen los 16 efectos que se pueden aplicar a una template de 1 pixel y que va f0.png, template que reproduce el vídeo sin ningún efecto, hasta el último <PIXMAP>1x1v01/ff.png</PIXMAP> que trata el vídeo reproduciendo solo el cuarto inferior derecha reflejado en caleidoscopio en las otras 3 partes. Puedes ver todos estos efectos en una tabla más adelante.

Cierra el conjunto el comentario <!-- 1x1 --> y que sirve para recordarnos que este pixgroup es de templates de 1 pixel.

## VIDEOBANKS y VIDEOBANK

```

130 <!-- list of grouped videos: root path from data/ -->
131 <!-- paths from BANKSPATH -->
132 <!-- OF_LOOP_NONE=1, OF_LOOP_PALINDROME=2, OF_LOOP_NORMAL=3 -->
133 <VIDEOBANKS BANKSPATH = "movies/banks/">
134 <VIDEOBANK> <!-- 1-->
135 <VIDEO SOURCEPATH="SOPA/_SOPA.mov" PLAYAUDIO="0" LOOPSTATE="3"></VIDEO>
136 </VIDEOBANK>
137 <VIDEOBANK> <!-- 2-->
138 <VIDEO SOURCEPATH="ANGURIAESPAGUETTILENGACARN/ANGURIA_1.mov" PLAYAUDIO="0" LOOPSTATE="3"></VIDEO>
139 <VIDEO SOURCEPATH="ANGURIAESPAGUETTILENGACARN/ANGURIA_2.mov" PLAYAUDIO="0" LOOPSTATE="3"></VIDEO>
140 <VIDEO SOURCEPATH="ANGURIAESPAGUETTILENGACARN/ANGURIA_ESPAGUETTI LENGACARN.mov" PLAYAUDIO="0" LOOPSTATE="3"></VIDEO>
141 <VIDEO SOURCEPATH="ANGURIAESPAGUETTILENGACARN/ANGURIA_TOT.mov" PLAYAUDIO="0" LOOPSTATE="3"></VIDEO>
142 <VIDEO SOURCEPATH="ANGURIAESPAGUETTILENGACARN/LENGACARN.mov" PLAYAUDIO="0" LOOPSTATE="3"></VIDEO>
143 <VIDEO SOURCEPATH="ANGURIAESPAGUETTILENGACARN/SPAGUETTI_1.mov" PLAYAUDIO="0" LOOPSTATE="3"></VIDEO>
144 <VIDEO SOURCEPATH="ANGURIAESPAGUETTILENGACARN/SPAGUETTI_2.mov" PLAYAUDIO="0" LOOPSTATE="3"></VIDEO>
145 <VIDEO SOURCEPATH="ANGURIAESPAGUETTILENGACARN/SPAGUETTI_TOT.mov" PLAYAUDIO="0" LOOPSTATE="3"></VIDEO>
146 </VIDEOBANK>

```

## VIDEOBANKS, VIDEOBANK y VIDEOS

En los primeros renglones vemos los siguientes comentarios:

<!-- list of grouped videos: root path from data/ --> lista de grupos de videos

<!-- paths from BANKSPATH -->

<!-- OF\_LOOP\_NONE=1, OF\_LOOP\_PALINDROME=2, OF\_LOOP\_NORMAL=3 -->

Instrucciones para reproducir el vídeo sin loop 1, con loop palindromo 2, loop normal 3.

En el siguiente renglón vemos

<VIDEOBANKS BANKSPATH = "movies/banks/">

VIDEOBANKS indica el conjunto de VIDEOBANK y BANKSPATH = "movies/banks/" que nos indica la situación de la carpeta banks en nuestro ejemplo [D:\...\pixMapsMicCam\data\](#).

Recuerda que la carpeta movies contiene además alfas y captures.

En el siguiente renglón vemos

<VIDEOBANK> <!-- 1-->

<VIDEO SOURCEPATH="SOPA/\_SOPA.mov" PLAYAUDIO="0"

LOOPSTATE="3"></VIDEO>

</VIDEOBANK>

el primer videobank, nos lo indica el comentario <!-- 1-->,

después el vídeo VIDEO SOURCEPATH con parte de la ruta SOPA para la carpeta, slash y SOPA.mov para el movie.

PLAYAUDIO="0" no reproduce el sonido y LOOPSTATE="3" ejecuta un loop normal, al llegar al final empieza de nuevo.

Observa que este videobank sólo contiene un vídeo pero en el siguiente, <VIDEOBANK> <!-- 2-->, tiene ocho. El orden en como situemos esta lista de vídeos, será el que utilizará, como veremos, la penúltima cifra hexadecimal del pixel correspondiente, el 0 del pixel será el primero de la lista, el 1 pixel el segundo de la lista, etc.

## ALFAVIDEOBANKS y ALFAVIDEOBANK

```

203
204 <ALFAVIDEOBANKS BANKSPATH = "movies/alfas/">
205 <ALFAVIDEOBANK>
206 <VIDEO SOURCEPATH="00_NAUOU_FRAME" PLAYAUDIO="0" LOOPSTATE="3"></VIDEO> <!-- 00 -->
207 <VIDEO SOURCEPATH="01_arribo_senjala" PLAYAUDIO="0" LOOPSTATE="3"></VIDEO>
208 <VIDEO SOURCEPATH="02_arribo_saluda" PLAYAUDIO="0" LOOPSTATE="3"></VIDEO>
209 <VIDEO SOURCEPATH="03_ultra_dream" PLAYAUDIO="0" LOOPSTATE="3"></VIDEO>
210 <!--
211 <VIDEO SOURCEPATH="test_alfa.mov" PLAYAUDIO="0" LOOPSTATE="3"></VIDEO>
212 <VIDEO SOURCEPATH="test_alfa2.mov" PLAYAUDIO="0" LOOPSTATE="3"></VIDEO>
213 -->
214 </ALFAVIDEOBANK>
215 </ALFAVIDEOBANKS>
216
217 </CONFIG>
    
```

### ALFAVIDEOBANKS, ALFAVIDEOBANK y secuencias de png alfa

ALFAVIDEOBANKS contiene de mismo modo que los anteriores conjuntos un número determinado de ALFAVIDEOBANK que en este ejemplo es sólo uno.

Las secuencias de png's se localizan en la lista que contiene el ALFAVIDEOBANK y son como esta:

```

<VIDEO SOURCEPATH="00_NAUOU_FRAME" PLAYAUDIO="0" LOOPSTATE="3"></VIDEO> <!-- 00 -->
-->
    
```

"00\_NAUOU\_FRAME" es el nombre de la carpeta que contiene la secuencia de png's dentro de la ruta del ordenador de gráficos

D:\...\pixMapsMicCam\data\movies\alfas

En nuestro ejemplo solo hay un ALFAVIDEOBANK pero podría haber varios.

PLAYAUDIO="0" nos indica que no activa el audio, al ser secuencia de png's tampoco se podría.

LOOPSTATE="3" nos indica que ejecutará un loop normal. La función palíndromo no es efectiva.

Las notas siguientes hacen referencia a una prueba anterior que ha quedado comentada:

```

<!--
<VIDEO SOURCEPATH="test_alfa.mov" PLAYAUDIO="0" LOOPSTATE="3"></VIDEO>
<VIDEO SOURCEPATH="test_alfa2.mov" PLAYAUDIO="0" LOOPSTATE="3"></VIDEO>
->
    
```

## CAPTUREPATH y VIDEOFOLDERS

```

97
98 <!-- folders with videos named outputX.mov with X = 0..15, paths from data/movies/captures -->
99 <CAPTUREPATH>movies/captures/</CAPTUREPATH> <!-- root path to save and load video captures -->
100 <VIDEOFOLDERS>
101 <VIDEOFOLDER>videos1/</VIDEOFOLDER>
102 </VIDEOFOLDERS>
103
    
```

CAPTUREPATH contiene el grupo de VIDEOFOLDERS. Los VIDEOFOLDERS tienen el nombre de videos1, videos2, etc. y contienen los vídeos de las capturas. Estas carpetas están situadas en la ruta:

## TEMPLATES. INTRODUCCIÓN

Las templates son el marco sobre el que se ejecuta todos los vídeos en PIXELMAPS a excepción de la cámara y de las animaciones.

ATENCIÓN trabajamos con píxeles RGBA, red green blue & alfa, si no son este tipo de píxeles la template no funciona. Existen otros tipos de píxeles como por ejemplo: blanco y negro, RGB, cmyk... estos no funcionan!!!

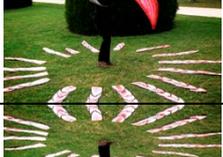
Para que una template de un pixel, 1x1, o un mosaico de pixeles, hasta 32x24, pueda ser interpretada por PIXMAPS tenemos que editar su número hexadecimal de ocho cifras [0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f].

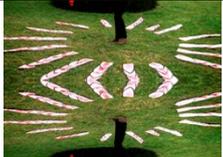
Tanto si editamos una plantilla de 1x1, como si creamos templates más complejas hay que tratarlas pixel a pixel. Cada pixel tiene los componentes red, green, blue, alpha y ocupan dos cifras cada componente siendo las dos últimas para el alfa. Así una cifra de 8 dígitos hexadecimal por ejemplo esta **7c2f2ca6**, se divide red 7c en green 2f, blue 2c, alpha a6. Al ser hexadecimal cada uno de estos componentes pares admite 256 valores (16x16), de 0 a 255. El programa PIXMAPS puede utilizar los 6 primeros dígitos de Red, Green, Blue, pero cuyo efecto solo sirve para tinter el vídeo que contiene, mientras que los dos últimos dígitos que corresponden al canal Alfa son los que realmente utilizamos. La penúltima cifra el número hexadecimal indica el número de un vídeo en una lista de hasta 16 vídeos de un solo bank. Recuerda que los Banks son las listas que confeccionamos en el documento **appSettings.xml** como hemos visto en el capítulo anterior. La última cifra nos permite aplicar un efecto al vídeo que hayamos elegido .

Ejemplo de número hexadecimal y su función en una template

Red	Red	Green	Green	Blue	Blue	Nº vídeo	Efecto
<b>7</b>	<b>c</b>	<b>2</b>	<b>f</b>	<b>2</b>	<b>c</b>	<b>a</b>	<b>6</b>

**TABLA HEXADecimal PARA CADA PIXEL DE UNA TEMPLATE EN PIXMAP**

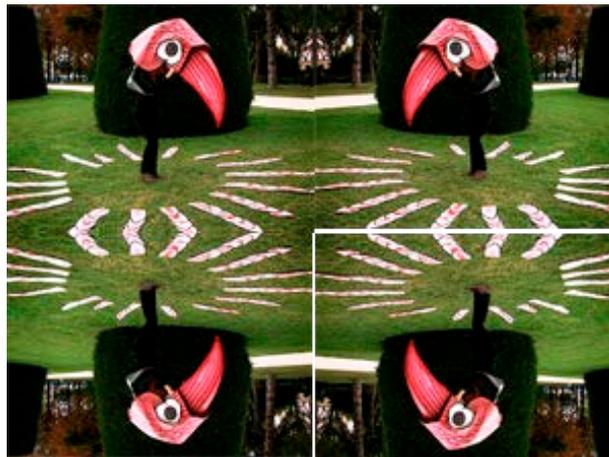
Cifra Tipo Decimal	Cifra Tipo Hexadecimal	Penúltima cifra / Nº video del bank	Última cifra / Efecto video	Ejemplo visual Efecto video
1	0	Video 1	El video se reproduce Normal	
2	1	Video 2	Vertical flip	
3	2	Video 3	Horizontal flip	
4	3	Video 4	Horiz/vertical flip	
5	4	Video 5	La mitad superior se reproduce en espejo en la inferior	
6	5	Video 6	La mitad izquierda se reproduce en espejo en la derecha	
7	6	Video 7	La mitad inferior se reproduce en espejo en la superior	
8	7	Video 8	La mitad derecha se reproduce en espejo en la izquierda	
9	8	Video 9	La mitad superior se reproduce abajo y su espejo arriba	
10	9	Video 10	La mitad izquierda se reproduce a la derecha y su espejo a la izquierda	

11	<b>A</b>	Video 11	La mitad inferior se reproduce arriba y su espejo abajo	
12	<b>B</b>	Video 12	La mitad derecha se reproduce a la izquierda y su espejo a la derecha	
13	<b>C</b>	Video 13	El cuarto superior izquierda se reproduce en calidoscopio en las otras 3 partes	
14	<b>D</b>	Video 14	El cuarto superior derecha se reproduce en calidoscopio en las otras 3 partes	
15	<b>E</b>	Video 15	El cuarto inferior izquierda se reproduce en calidoscopio en las otras 3 partes	
16	<b>F</b>	Video 16	El cuarto inferior derecha se reproduce en calidoscopio en las otras 3 partes	

### TEMPLATES MOSAICO

Como hemos dicho programa PIXMAPS admite templates de 1x1 pixel hasta 32x24 pixel. Cualquier formato intermedio por ejemplo 16x1 es admitido y ajustará cada uno de los pixeles/vídeos al formato rectangular de la proyección. Veamos pues sobre el frame del vídeo tomado como ejemplo anteriormente y que es el primer video de un VIDEOBANK y por tanto el número 0 en la penúltima cifra hexadecimal de los cuatro pixeles que conforman una template de 2x2 pixeles.

0 2 horizontal flip	0 0 normal
03 horizontal/vertical flip	0 1 vertical flip



Otro ejemplo 1x3 con el mismo vídeo

0 2 horizontal flip	0 5 La mitad izquierda se reproduce en espejo en la derecha	0 0 normal
------------------------	---	---------------



## **CREAR LAS TEMPLATES con el programa INKSCAPE.**

Para editar el número hexadecimal de cada uno de los píxeles que utilizemos para construir las templates necesitaremos un editor de gráficos capaz de cambiar dicho número. Os proponemos en este manual Inkscape un editor Open Source.

Inkscape es un editor de gráficos vectoriales gratuito y de código libre. Inkscape puede crear y editar diagramas, líneas, gráficos, logotipos, e ilustraciones complejas. El formato principal que utiliza el programa es Scalable Vector Graphics (SVG) versión 1.1.

Inkscape tiene como objetivo proporcionar a los usuarios una herramienta libre de código abierto de elaboración de gráficos en formato vectorial escalable (SVG) que cumpla completamente con los estándares XML, SVG y CSS2.

Inkscape se encuentra desarrollado principalmente para el sistema operativo GNU/Linux, pero es una herramienta multiplataforma y funciona en Windows, Mac OS X, y otros sistemas derivados de Unix.

Es una aplicación disponible en muchas lenguas, incluyendo sistemas de escritura complejos (como sistemas de escritura de derecha a izquierda como árabe, hebreo...).

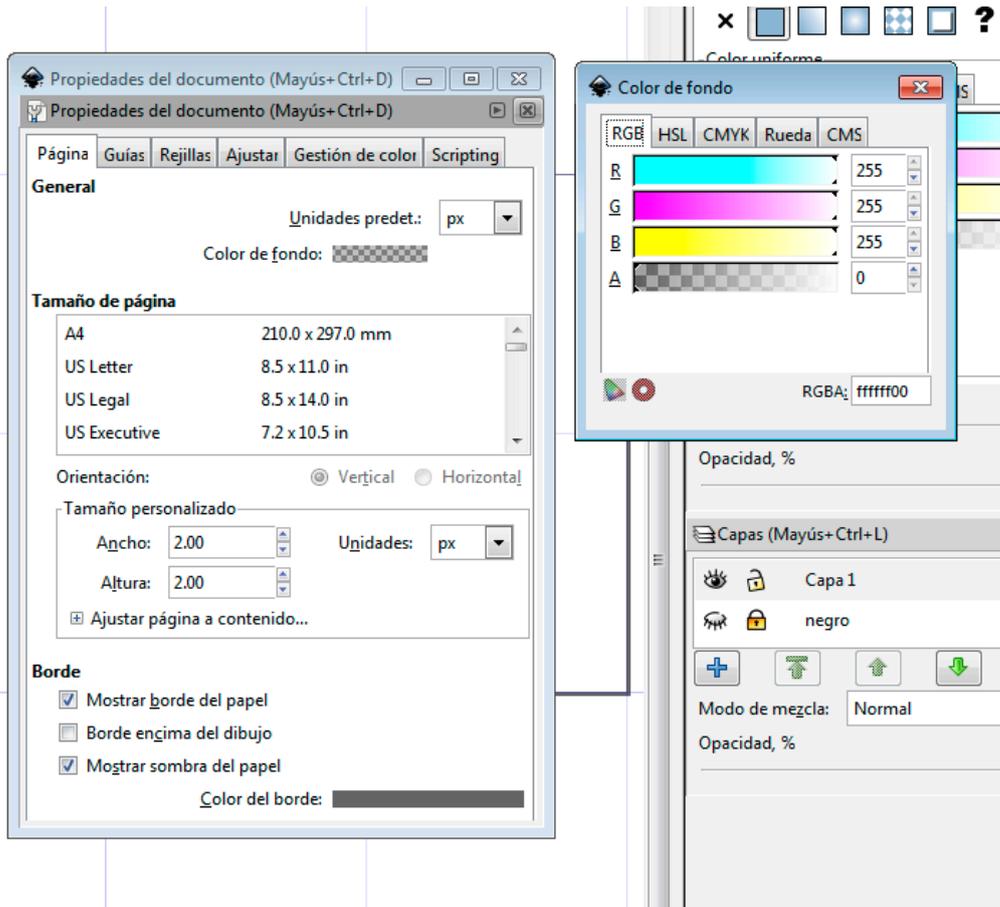
Este programa surgió de una bifurcación del proyecto Sodipodi.

Hacemos las templates con Inkscape, es un programa open source que podéis descargar en:

<http://inkscape.org/download/>

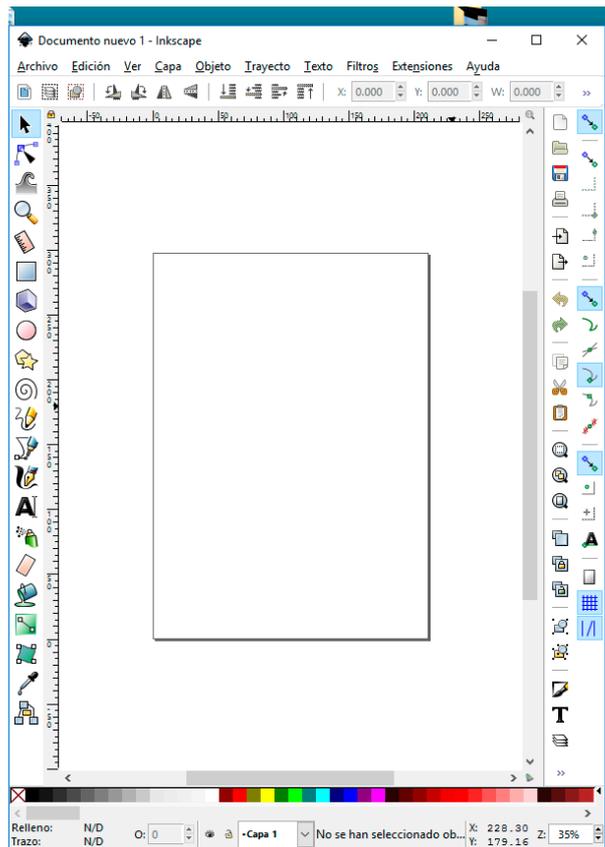
<b>OJO MUY IMPORTANTE NO CUALQUIER PNG NOS SIRVE SOLO LOS PÍXELES RGBA</b>
--

Lo primero que hay que hacer es configurar el color de las propiedades del documento. Esto está en el menú archivo/propiedades del documento, en la casilla color de fondo (doble clic) R255, G255, B255, A 0; que en la casilla hexadecimal es ffffff00.

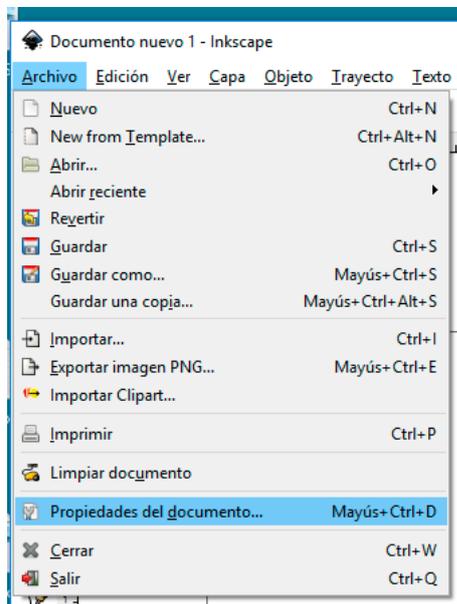


Para crear una platilla nueva:

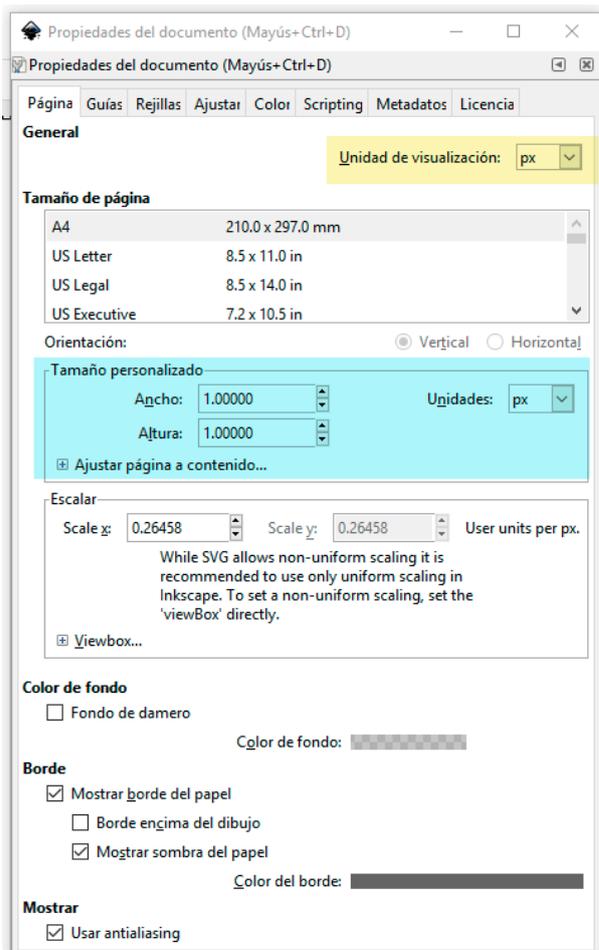
1º abrir el programa, se supone que ya hemos elegido el número de pixeles/video que tendrá nuestra matriz, por ejemplo 1 x 1, pero por defecto aparece un documento más grande:



Iremos al menú archivo / propiedades del documento/ (señalado en la imagen en azul)



Se abrirá el submenú propiedades del documento. Poner la Unidad de visualización (señalada en amarillo) en píxeles. Cambiar el tamaño personalizado (señalado en azul) a un píxel, o al número de píxeles que queramos hacer la template, al escribir 1 en la casilla de altura para que el documento se convierta en 1x1 dar al enter, observareis que el documento se hace pequeño:



Es probable que sea tan pequeño que no veáis el documento. Para hacerlo más grande en el escritorio del Inkscape hay varias maneras, yo utilizo la tecla control + ruedecilla del mouse. Si os desaparece el documento al engrandecerlo podéis darle a Control + b, aparecerán los sliders en la base y la izquierda del escritorio, con ellos podéis buscar la pagina

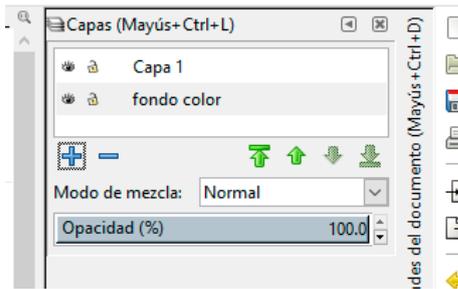
Puedes montarte una capa de fondo de color o en negro para ver el alfa de los pixeles. Esto facilita ver el pixel genérico que hemos determinado ya que es transparente de modo que los cambios que hagamos no los veremos si no tenemos una capa de fondo.

Para ver las capas tenéis que ir al menú capa, y clicando la última línea Capas... os aparecerá,

un submenú que contiene este Icono: 

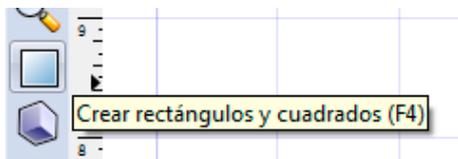
hay una atajo: ctrl+shift+L.

Con este botón añades una capa, en este caso la llamamos "fondo color" y con la casilla de un nuevo submenú **posición/debajo de la actual**, o con la flecha que indica hacia abajo la movemos a la parte inferior, etc.



Observad que las capas tienen un ojo y un candado   que sirven para hacer invisible la capa y para bloquearla. Esta herramienta os va a ser muy útil en el momento de guardar como en \*.png, se trata de bloquear la capa que utilizéis de fondo y de este modo el \*.png solo guardara la capa sobre la que hemos trabajado los pixeles.

Con f4 o la herramienta de "crear rectángulos y cuadrados"



que está en el menú de la derecha del programa, seleccionáis la capa de fondo. Y quedará de un color, para controlar el color tenéis que abrir el menú Objeto y dentro submenú de relleno y borde (shift+Ctrl+F) aparece el submenú del RGBAlfa que os permite colorear esa selección, yo la coloreo la capa fondo en negro y la bloqueo. Esta herramienta también me sirve para añadir los pixeles de capa 1, con la que trabajaré el transcriptor que hemos apuntado antes. Se trata en esta capa de ir creando pixeles con las características, que apunto en el capítulo anterior.

Como hemos visto la aplicación PixMaps utiliza la numeración hexadecimal. El lugar donde nos podemos ver la cifra hexadecimal está en la casilla RGBA de parte inferior izquierda del submenú de Borde y Relleno



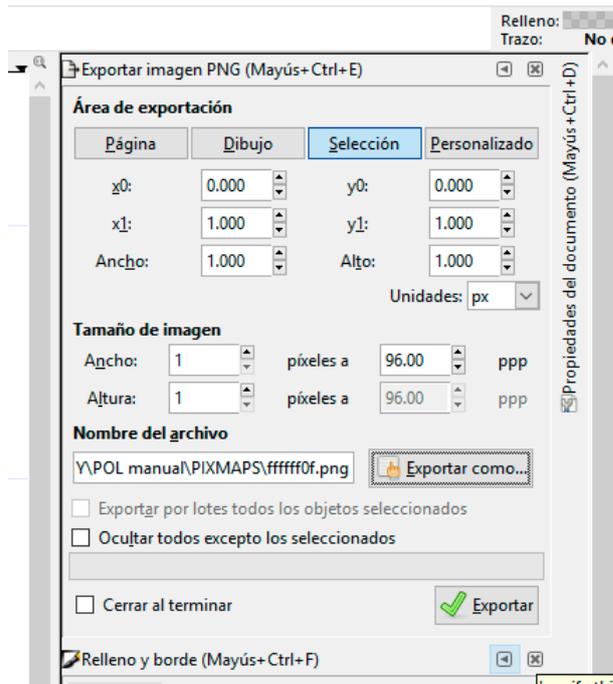
Esta casilla permite insertar las cifras a mano. Allí deberemos poner el número de video/captura = penúltima cifra, y el efecto= última cifra, es decir los dígitos correspondientes al ALPHA. Ver tabla pág. 4.

COMO GUARDAR LAS TEMPLATES.

ATENCIÓN LOS PNG NO SE GUARDAN desde **guarda como** si no desde el **archivo/Exportar mapa de bits...** Allí se abre el submenú Exportar mapa de bits (mayus+ctrl+E). De este submenú nos interesan:

**Área de exportación.** Tenemos que seleccionar los píxeles que queremos exportar de la capa de trabajo y bloquear la capa de fondo que hemos utilizado para ver mejor la capa de trabajo. En las casillas nos aparecerá el tamaño del documento que exportamos. En la imagen estamos exportando un documento de 1 píxeles 1x1.

**Nombre del archivo.** Nos permite determinar la ruta donde vamos a guardar el documento y su nombre. Después dale al botón **Exportar**



A parte de guardar los \*.png de esta forma es aconsejable crear 2 carpetas cuando trabajes las plantilla/templates. En una guardas los \*.svg que es el formato que utiliza Inkscape, y que mantiene todas las propiedades del documento y en otra carpeta exportáis los \*.png como mapas de bits, que como ya he dicho son las verdaderos PIXMAPS templates. De modo que si ves que los \*.png no funcionan puedes volver al archivo \*.svg y modificarlo.

**TABLA DE MENSAJES OSC EN PIXELMAPS**

Mensajes OSC para manejar PIXELMAPS desde POL o desde otro programa.

## Generales

mensaje	valor	descripción
/debug		Para ver los mensajes de PIXELMAPS en pantalla funciona como toogle, cambia según el estado anterior
/fullscreen		Por defecto PIXELMAPS se reproduce en fullscreen, si ponemos este mensaje sacara el fullscreen , funciona como toogle
showCamera	si	Con 1 pone la cámara sobre pixmaps, con 0 la saca
showCamera/mode	si	Con 1 audio   con 2 fade, hace fade de 0 a 100 al /value
/showMicCam/value	si	Fade sobre la cámara de 0 a 100

Para reproducir los vídeos y las capturas se necesita siempre de una template. Puede ser para un sólo vídeo, en este caso el template será de 1 pixel o de varios pixeles con lo que se crea un mosaico de videos. Tanto si es de un pixel como de varios la template contiene la información para definir efecto, color y número de video.

## TEMPLATES y GROUPS mensajes

mensaje	valor	descripción
/playPixMap/visible 1	si	Muestra PIXMAPS con el valor 1 , con 0 lo oculta
/playPixMap/close 1	no	Cierra PIXMAPS al completo
/playPixMap/group	si	Con este mensaje elige PIXGROUP y con el número el PIXGROUP específico según su posición en la lista PIXMAPS
/playPixMap/group/next	no	Próximo PIXGROUP de la lista que contiene los contiene todos PIXMAPS
/playPixMap/group/prev	no	Anterior PIXGROUP de la lista que contiene los contiene todos PIXMAPS
/playPixMap/tmp	si	Carga el template llamando a un número de la lista de PIXMAPS del PIXGROUP actual determinado
/playPixMap/tmp/next	no	Próximo template PIXMAP del un PIXGROUP actual
/playPixMap/tmp/prev	no	Anterior template PIXMAP del PIXGROUP actual

## CAPTURAS Mensajes

/playPixMap/dir	si	Elije una carpeta de capturas de la lista VIDEOFOLDER del conjunto VIDEOFOLDERS. La captura es el número específico según su posición en la lista del VIDEOFOLDER
/playPixMap/dir/next	no	Próxima carpeta de capturas de la lista VIDEOFOLDER
/playPixMap/dir/prev	no	Anterior carpeta capturas de la lista VIDEOFOLDER

VIDEOS y BANKS mensajes

/playPixMap/bank	si	Elije un banco de vídeos de la lista VIDEOBANK del conjunto VIDEOBANKS. El número específico es según su posición en la lista del VIDEOBANK.
/playPixMap/bank/next	no	Próximo VIDEOBANK de la lista de VIDEOBANKS
/playPixMap/bank/prev	no	Anterior VIDEOBANK de la lista de VIDEOBANKS

ALFAS secuencias de png's Mensajes

/playAlfa/bank/close	no	Cierra la funcionalidad de secuencias con alpha
/playAlfa/bank/video	si	Carga una carpeta de secuencias de un ALFAVIDEOBANK determinado, indicándole su número en posición
/playAlfa/bank	si	Elije un ALFAVIDEOBANK de todos los que contiene ALFAVIDEOBANKS

CONTROL VIDEO Mensajes

/playPixMap/video/pause	si	(toogle)
/playPixMap/video/speed F	si	0 pausa, 1 normal, -1 reverse, 2 doble velocidad, -2 reverse doble velocidad
/playPixMap/video/frame F	si	Nº de frame
/playPixMap/video/seek F	si	Nº de frame
/playPixMap/video/position F (%)	si	
/playPixMap/video/showCamera	si	Muestra el numero de la cámara en una lista de cámaras
/playPixMapTop/	no	[same as playpixmap] para poner un 2º template/video encima de otro

## Teclado

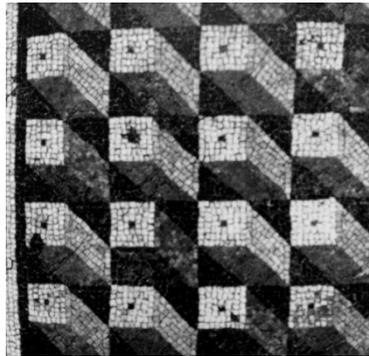
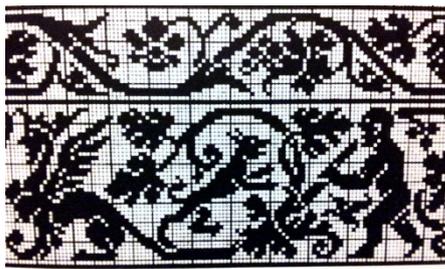
Desde el teclado del ordenador de gráficos donde está instalado PIXELMAPS puedes activar algunas de las órdenes anteriores

d	Debug
f	Fullscreen
UP	next template
DOWN	previous template
PAGE_UP	previous group
PAGE_DOWN	next group
HOME	previous videobank
END	next videobank
RIGHT	next folder
LEFT	previous folder
F1	change Pix Template 0
F2	change Pix Template 1
F3	change Pix Template 2
a	previous ALFAVIDEOBANK
z	next ALFAVIDEOBANK
i	start capture game
r	record video capture
s	stop video capture
e	delete last video capture
o	stop capture game

## ANEXO A PIXMAPS Y TEMPLATES

Utilizamos la técnica de las templates/mosaico en la performance PSEUDO en las escenas de Duccio Doccia y en Mara Mara. Las templates/mosaico video son utilizadas en estas escenas como fondos de animaciones flash. En esta performance PIXMAPS tenía activo el sistema de capturas de modo que los directorios que utilizaban los distintos pixeles eran capturas realizadas durante la performance.

La idea de utilizar estas templates está inspirada en tapices y mosaicos como los siguientes:



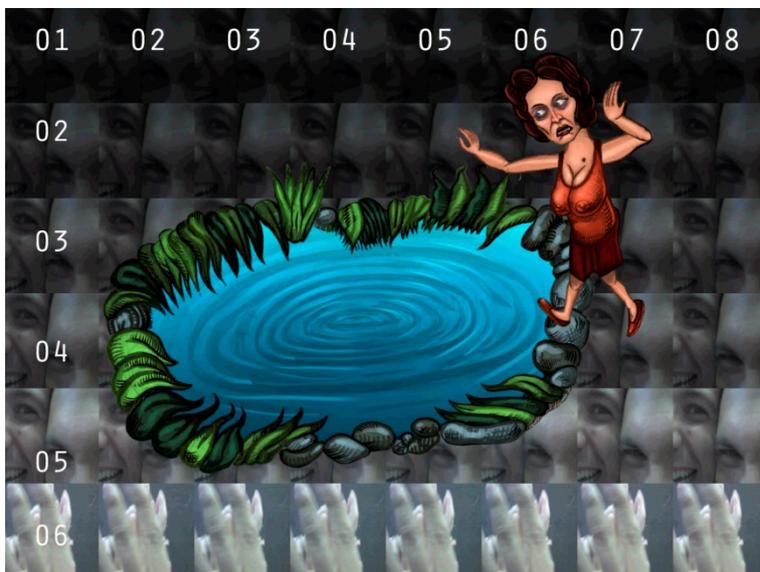
La estrategia de interacción en PSEUDO permite situar los videos que acabas de grabar en una parte anterior dentro de una platilla previamente preparada. Es decir los videos son cargados en cada uno de los pixeles de la imagen.

A continuación una plantilla de 32 por 24 pixeles, que carga distintos videos y distintas orientaciones, observad los pixeles del ojo que algunos tienen un flip vertical.

Esta imagen y las siguientes tienen una animación con el fondo en alfa hecha en flash, de modo que el POL display manager sitúa la plantilla de pixeles/video al fondo y el swf de flash en la parte superior. En la versión de PIXMAPS que acabamos describimos en este manual las animaciones podría estar realizadas con los ALFAVIDEOBANKS.



A continuación una imagen de 8 por 6 pixeles, en esta imagen permite reconocer mejor los videos y tienen más importancia, no obstante mantiene un tinte en gris dada línea para dar la sensación de profundidad. Observad que las líneas verticales del 1 la 5 van degradando el tinte hasta llegar a la 6 a la línea sin tinte.

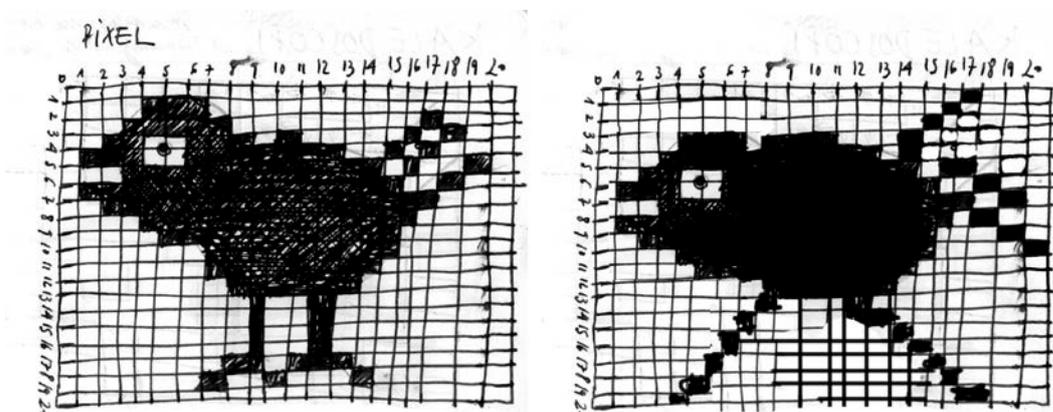


Por último incluyo la imagen de una parte de la escena Mara Mara, también de PSEUDO, en la que se utilizan solo 4 pixeles 2 videos. Aquí se me ocurre que según las imágenes que graves se puede trabajar con la idea de calidoscopio, simplemente disponiendo de un único video en cuatro posiciones: Normal, flip horizontal, flip vertical, flip vertical/horizontal.

Esta es una matriz de 4 píxeles utilizada en la escena MARA MARA de PSEUDO



Como funciona en realidad, es flash que le dice a través de POL que imagen tiene que cargar, de modo que es posible cargar varias plantillas durante una misma parte e incluso en un breve espacio de tiempo. Esta opción no se hizo en PSEUDO pero la vamos a hacer en este taller, mi idea es que preparemos plantillas OF tratándolas como frames de una animación. Desde el programa POL podemos llamar estas plantillas en forma de secuencia rápida y crear el efecto de animación. Un ejemplo, hecho a mano de 2 frames,



Si planteáis una secuencia más compleja podréis conseguir efectos de animación más interesantes.

A estas alturas del escrito ya habréis entendido que hay dos factores importantes en este ejercicio, uno depende del tipo de video que grabas, así para conseguir el efecto de pixel oscuro y pixel claro de la imagen anterior deberás grabar una película oscura y otra clara y utilizar solo estos dos videos. El otro factor depende de cómo se haga la plantilla/template, en el caso anterior está muy claro la disposición de píxeles crea efecto de movimiento.