

POL_MATRIXv1.pol.

INDICE

POL_MATRIXv1.pol. DESCRIPCION DE LA PERFORMANCE	2
COPIAR LAS RULES	2
PIXMAPS Y POL SETUP	2
TABLA DE MENSAJES OSC EN PIXELMAPS	3
TABLA MENSAJES POL	5
POL_MATRIXv1.pol. Scenes y Rules.	5
000 CAMERA.	6
001 VIDEO_PIXMAPS SCENE	9
002 BANK_VIDEO+PIXMAPS SCENE.	13
003 DIR+TEMPLATE+GROUPS SCENE.	17
004 CAMERA+ALFAVIDEObANK SCENE	20
005 BANK+TEMPLATES+SEQ_PNG SCENE.	25
006 FLASH SCENE	30
TABLA VARIABLES para CONTROL del REPRODUCTOR de FLV en FLASH	32
007 PART CHANGING.	40
008 SCENE. POL COUNTER VARIABLES	47
009 SCENE. POL COUNTER VARIABLES EXAMPLE VIDEO.	53
010 LIGHTS CONTROL (D-LIGHT)	53

POL_MATRIXv1.pol. DESCRIPCION DE LA PERFORMANCE

Este ejemplo nos sirve para entender de qué manera funciona POL Setup. Puedes descargar el archivo el zip de POL_MATRIXv1.pol en el site <https://systorgy.hangar.org> En él tienes los ejemplos de movies, XML, animaciones, templates necesarias para ejecutar esta performance. Sustituyendo estos archivos, retocando los XML y sustituyendo las IP necesarias puedes hacer tu propia performance. En este ejemplo hemos escrito varias scenes (partes) solo es un pequeño ejemplo, no están todas las posibilidades que ofrece el programa que son muchas. Con el manual de POL Setup puedes integrar las nuevas scenes y rules que necesites. Verás que en el zip hay dos performances la que se ejecuta en una sola maquina (LOCAL) y la que utiliza un sistema distribuido de varios ordenadores que es la que utilizamos nosotros y que tiene nuestras IP y el nombre de nuestros equipos, esta es la que utilizamos para explicar este ejemplo. Verás que la diferencia entre una y otra reside en cómo está configurado el panel Devices Configuration, de POL Setup. Desde allí puedes, configurar las nuevas rutas, cambiar las IP y los nombres de los equipos.

Como puedes ver en el manual de POL Setup esta aplicación permite mapear distintos dispositivos físicos y virtuales conectándolos entre sí. Permite también crear un conjunto de reglas interactivas y/o automáticas entre Interfaz y Médium que se organizan en archivos y que tienen raíz *.pol. A estos archivos les llamamos **performance**. Las performances se estructuran en partes que en POL llamamos **scenes**. La scenes sirven para organizar el tiempo, la carga de nuevos contenidos y/o de nuevas situaciones interactivas. Las scenes están formadas por reglas, llamadas aquí **rules**, que permiten agrupar, como veremos, distintas acciones que llamamos **events**. La rules permiten cargar un nuevo contenido, un *.mov y/o un *.wav por ejemplo, establecer los sensores con los cuales vamos a alterar los contenidos, cargar una memoria iluminación, etc.

Las rules establecen relaciones entre **source** y **target**. Las **source** son las fuentes de entrada, ya sea una interfaz, un canal, una aplicación o el propio programa POL. Los **target** son los distintos canales que nos permiten actuar sobre otras aplicaciones o dispositivos físicos o el propio programa POL. Los target acaban teniendo consecuencias en las aplicaciones que gestionan, entre otras, los gráficos, el sonido, la iluminación o los robots.

COPIAR LAS RULES

Hemos escrito este ejemplo para que resulte fácil copiar sus rules y utilizarlas de nuevo. Las rules pueden ser fácilmente copiadas de una scene a otra basta con hacer copy de las rules que te interesen en una scene y paste en la scene de destino. Para copiar de una performance a otra: copia las rules, [elige las que quieras, selecciónalas y haz control+c], cierra la performance y sin cerrar POL Setup abre la nueva performance y copiar las rules [control+v] en la scene que desees. No guarda las rules portapapeles fuera de POL de modo que si cierras el programa pierdes la copia. Si tienes dos POL Setup abiertos no puedes pasar las rules de uno a otro programa.

Casi siempre cuando nos referimos a una información externa a este manual nos referimos a **POL Setup Manual**. El pdf de este manual esta en Systorgy. Hangar.org.

PIXMAPS Y POL SETUP

En esta performance POL se relaciona con la aplicación **PIXMAPS**.

A continuación, entre corchetes, algunos elementos de la nomenclatura de los mensajes de OSC de PIXMAPS que utiliza POL para comunicarse entre sí.

PIXMAPS utiliza los pixeles hexadecimales, 1x1, o sus mosaicos, de 1x2 hasta 32x24, como **Templates [tmp]** para contener los vídeos y las capturas de vídeos. Los conjuntos de Templates se organizan en **grupos [groups]**. Los **vídeos [video]** se sitúan en carpetas que llamamos **[videobank]**. No utilizamos los pixeles para contener la cámara ni las animaciones. Las animaciones están basadas en secuencias de png, y organizadas en carpetas **[alfa/bank]**. 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**].

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	0	Mensaje para cargar las animaciones, Imprescindible siempre para llamarlas. Siempre con valor 0
/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

CAPTURAS, Mensajes para la capturadora de video. (No disponible en esta versión)

/stateCapture		
/delVideoCapture		
/recVideoCapture		

Para configurar los groups, las templates y los videobanks lo hacemos desde el archivo XML de

appSettings.xml de PIXMAPS que se encuentra en:
D:\grafic_performances\systorgy\pixMapsMicCam\data

D: para el dispositivo de gráficos. **grafic_performances** para la carpeta que contiene las distintas carpetas raíz con contenidos de diferentes performances. **systorgy** para la carpeta que contiene los contenidos de esta performance. **PixMapsMicCam** para todos los contenidos, las DLL y exe de PIXMAPS. **data** para los contenidos, templates, movies, png's.
POL_MATRIXv1.pol. Scenes y Rules

Para más información consulta el **PIXMAPS manual** que encontrarás en: Systorgy.hangar.org

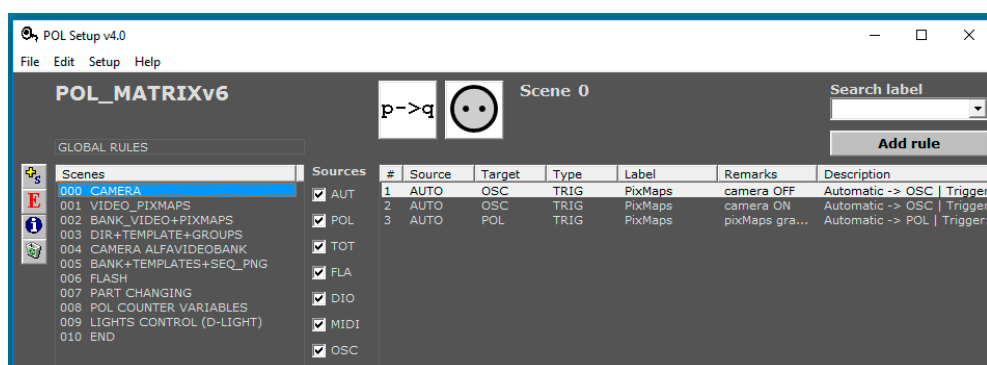
TABLA MENSAJES POL

POL tiene una serie de mensajes internos que permiten enviar órdenes al propio programa. Se refiere a mensajes concretos que suceden dentro del propio programa como cambiar de scene (Next Part), etc. Esta es la tabla y sus significados.

Next Part	Te lleva a la próxima Scene
Previous Part	Te lleva a la anterior Scene
Switch Player to App	Lleva la aplicación de gráficos al TOP de la pantalla gracias al POL Display Manager.
Enable Event	Activa un Rule
Disable Event	Desactiva una Rule
Increment Variable	Permite incrementar el counter de POL
Decrement Variable	Permite Decrementar el counter de POL
Assing Value to Variable	Permite asignar una variable concreta al counter de POL

POL_MATRIXv1.pol. Scenes y Rules.

Al abrir la performance en **POL Setup** aparece esta pantalla:



La columna de la izquierda contiene 11 scenes desde la 000 CAMERA hasta la 010 END. Observa que 000 CAMERA en blanco esta seleccionada. En la columna de la derecha contiene tres filas que corresponden a las tres rules de esta scene que veremos en detalle más adelante. Cada una de estas filas está organizada en siete columnas, que corresponden:

#. Numero de la rule

Source. Indica el canal de entrada, source, de la rule

Target. Indica el canal de salida, target, de la rule.

Type. El tipo de comportamiento

Label. La etiqueta que hemos asignado

Remarks. La nota que hemos escrito sobre la rule

Description. Propiamente la rule.

Esta rules deberían servirnos para entender cómo funciona su escritura y para copiarlas en una nueva performance, si así lo deseamos.

Para entrar en el submenú de una rule ya escrita tienes que darle doble clic sobre la misma y se te abrirá. Tal como se describió en detalle en el POL Setup Manual este submenú esta organizado en tres ámbitos, las casillas de la derecha, las de la izquierda y el espacio inferior que ocupa algo menos de la mitad de espació.

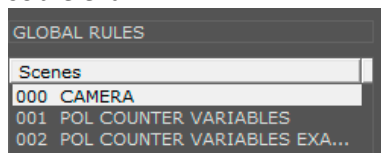
El ámbito de la izquierda, **sources**, nos sirve para determinar los dispositivos de entrada como las interfaces, los programas o la propia aplicación POL.

El ámbito de la derecha, **tagets**, nos sirve para definir sobre qué vamos a actuar y de que forma.

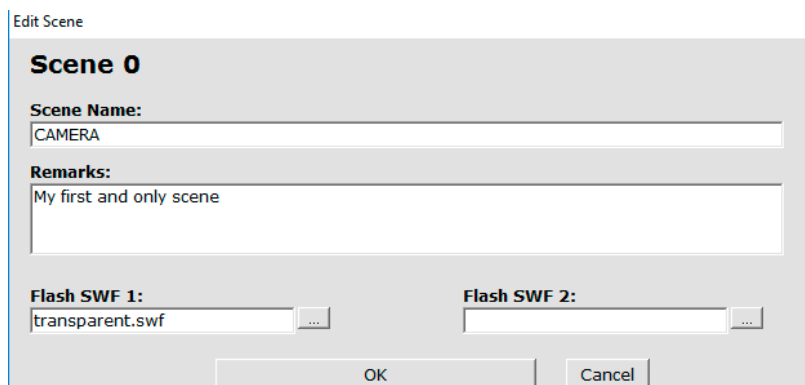
El ámbito de la mitad inferior son casillas que nos sirven para darnos información y facilitar la localización de la rule. El recuadro de **Actions List** nos deja ver lo/s even/s que hemos configurado en los ámbitos anteriores. En el caso que la rule contenga varios evens y quieras cambiar algo, tendrás que seleccionar el even y cambiarlo. En el caso que dentro de una rule ya escrita quieras añadir otro even, tendrás que copiar uno de los evens que tengas escrito y configurar esta nueva copia. En **remarks** escribimos la nota de la función de esta rule y en la casilla **label** le hemos asignado PixMaps. Puesto que este ámbito mitad inferior es meramente organizativo e informativo o describiremos las casillas

000 CAMERA.

En esta scene están el conjunto de rules que nos permiten cargar automáticamente una cámara IP y verla en pantalla al entrar en ella. Como puedes ver la scene esta seleccionada y aparece en blanco en la casilla de las Scenes, para seleccionarla simplemente hemos clicado sobre ella:



En esta performance ya nos vienen configurados todos los submenús de Scene. Si quieres cambiar algo de este submenú puedes acceder con doble click al nombre 000 CAMERA. En esta es la imagen del submenú edit scene es la imagen siguiente:



Como ya comentamos en la parte del POL Setup Manual, **EDIT SCENE / SCENES. COMO**

CREARLAS Y ORDENARLAS, este submenú tiene la casilla Scene Name, en la que hemos escrito el nombre CAMERA. Remarks en la que por defecto aparece esta frase: **My first and only scene**, la podemos borrar y escribir la nota que nos convenga para reseñar lo que nos parezca importante. Vemos también dos casillas Flash SWF. Recuerda que la primera en esta performance es el target de gráficos Flash, y el segundo es la tablet Android. Así pues en el primer pondremos **transparent.swf**. Se trata de una movie completamente transparente, y que permite ver aunque esté presente el otro target de gráficos PIXMAPS. Esta movie solo está como previsión en caso que no activemos el Pol Display Manager, como ya sabes este programa se encarga de llevar adelante la aplicación de gráficos que en nuestro caso son PIXMAPS y Flash. El **transparent.swf** necesario si no utilizamos flash.

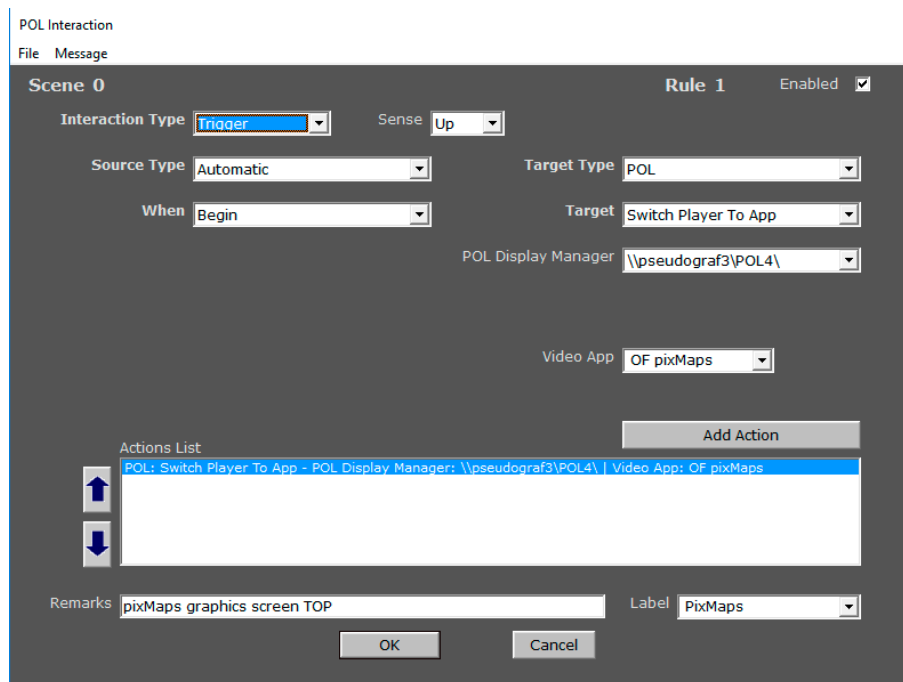
La scene CAMERA tiene tres rules tantas como filas.

#	Source	Target	Type	Label	Remarks	Description
1	AUTO	POL	TRIG	PixMaps	pixMaps graphics screen TOP	Automatic -> POL Trigger: Up Begin POL
2	AUTO	OSC	TRIG	PixMaps	camera ON	Automatic -> OSC Trigger: Up Begin OSC
3	AUTO	OSC	TRIG	PixMaps	camera OFF	Automatic -> OSC Trigger: Up End OSC

1 Rule. 000 CAMERA. pixMaps graphics Screen TOP.

Como llevar una aplicación de gráficos con PDM al primer término de la pantalla.

En esta rule el programa Pol Display Manager, PDM, situado en el ordenador de gráficos, manda arriba la aplicación de Pixmaps que queda en primer término del ordenador de gráficos. Aunque según el orden de las scenes esta rule puede resultar innecesaria, aconsejamos ponerla siempre.



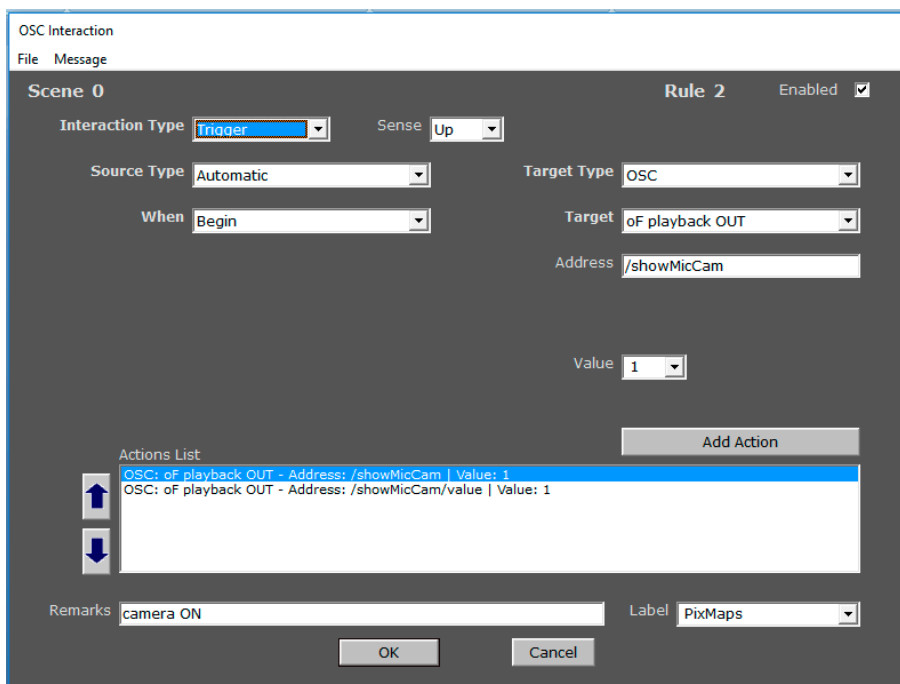
Las casillas de Source están igual que en las anteriores rules. El **target type** es aquí **POL** se refiere por tanto que es una orden que damos al propio programa. El **target** es **Switch Player to App** que como vimos permite que la aplicación de gráficos este en el TOP de la pantalla gracias al POL Display Manager. Vemos que la casilla siguiente ha cambiado de nombre y se llama **POL Display Manager** y que nos muestra el nombre del equipo gráficos, en nuestro ejemplo **\pseudograf3\POL4\'**, si ejecutas el programa en local saldrá la ruta donde tengas el ejecutable de Pol Display Manager. Si despliegas esta casilla observarás que también hay otro

target que dice **All Display Managers**, y se refiere al caso de que hubiera más de una equipo de gráficos poder mandar la orden a todos ellos. La casilla siguiente aquí también con un nuevo nombre Video App contiene Flash y OF pixMaps, las dos aplicaciones de gráficos que hemos configurado en el panel DC. El target **Switch Player to App** situa **OF pixMaps**, PIXMAPS, en primer término

Las casillas del menú que siguen a continuación en la parte baja de este panel se describieron en detalle en el POL Setup Manual y sirven para darnos información y facilitar la localización de la rule. El recuadro de Actions List nos deja ver lo/s even/s es este caso es solo uno. En remarks hemos escrito la nota de la función de esta rule y en la casilla label le hemos asignado PixMaps. No las describiremos en las rules siguientes.

2ª Rule. 000 CAMERA. Camera ON.

Como activar la cámara en la lista de cámaras del documento de XLM PIXMAPS.

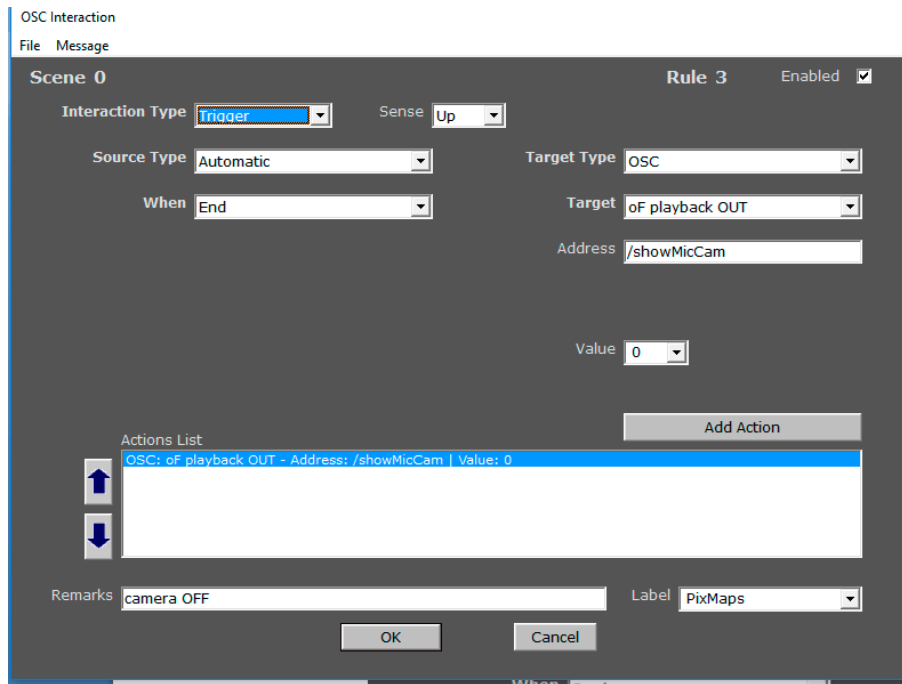


En la actions list hay dos events, si los seleccionas uno a uno verás que las Interaction Type, Sense y Source type están igual que en la anterior rule. El When está en Begin, esto indica que el target se activará automáticamente al entrar en esta parte. Observa que el Target Type y el Target están igual que en la anterior rule en ambos events. Vemos que la diferencia está en el address uno **/showMicCam, value 1**, activa la función de cámara, el otro **/showMicCam/value, value 1**, activa la cámara nº 1 de la lista de PIXMAPS.

3ª Rule. 000 CAMERA. Camera OFF.

Como desactivar la cámara PIXMAPS del sistema

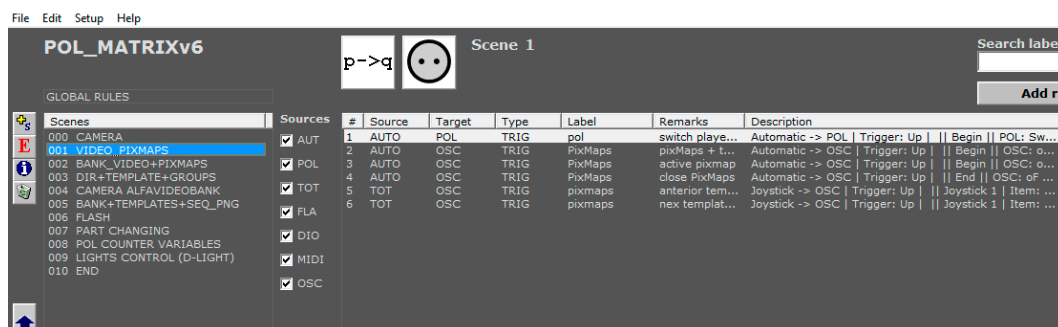
Si hacemos un doble clic sobre esta rule aparece el menú:



Observamos que la **Interaction Type** esta en **Trigger**, por tanto activara el target sin condiciones. El **sense** esta en **Up**, salvo excepciones se usa así. La **Source Type** está en **Automatic** esto significa que funcionará al entrar o al salir en la parte. El **When** está en **End** el Target ocurrirá al acabar al parte. Puesto que nos comunicamos a través de PIXMAPS la **Target type** es **OSC**. Su **target** es **oF playback OUT**, recordad que así de configuró en el panel Devices Configuration. La **Address** es **/showMicCam**, en la descripción de el programa PIXMAPS encontrareis la tabla de Address, para cada una de las opciones que ofrece este programa. La **Value** es **0** lo que nos indica en este caso que la Address esta inactiva y que por tanto la cámara se apagar al salir de la parte.

001 VIDEO_PIXMAPS SCENE

Cómo actuar con sensores ON/OFF sobre un vídeo, VIDEOBANK, al que le aplicamos distintas plantilla, TEMPLATES, que producen distintos efectos de espejo y calidoscopio.

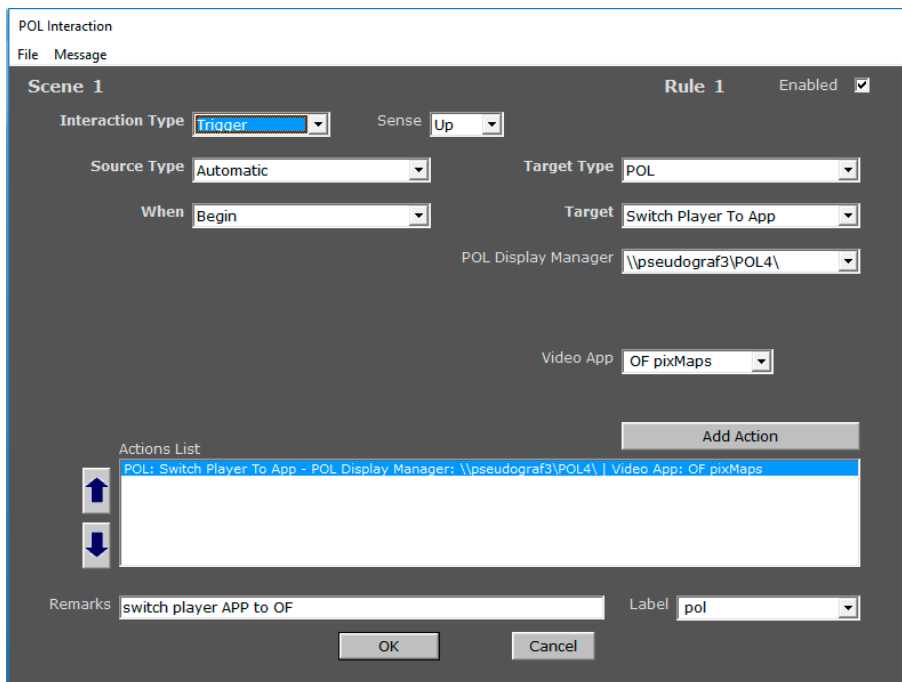


En esta escena contiene 6 rules que veremos a continuación. Esta rule como la anterior utiliza PIXMAPS como target.

01 rule 001 SCENE.

Ponemos automáticamente al cargar la scene PIXMAPS al frente gracias a POL Display Manager.

Revisa la informaciones de : 1 Rule. 000 CAMERA. pixMaps graphics Screen TOP. Esta rule es la misma.



02 rule 001 SCENE.

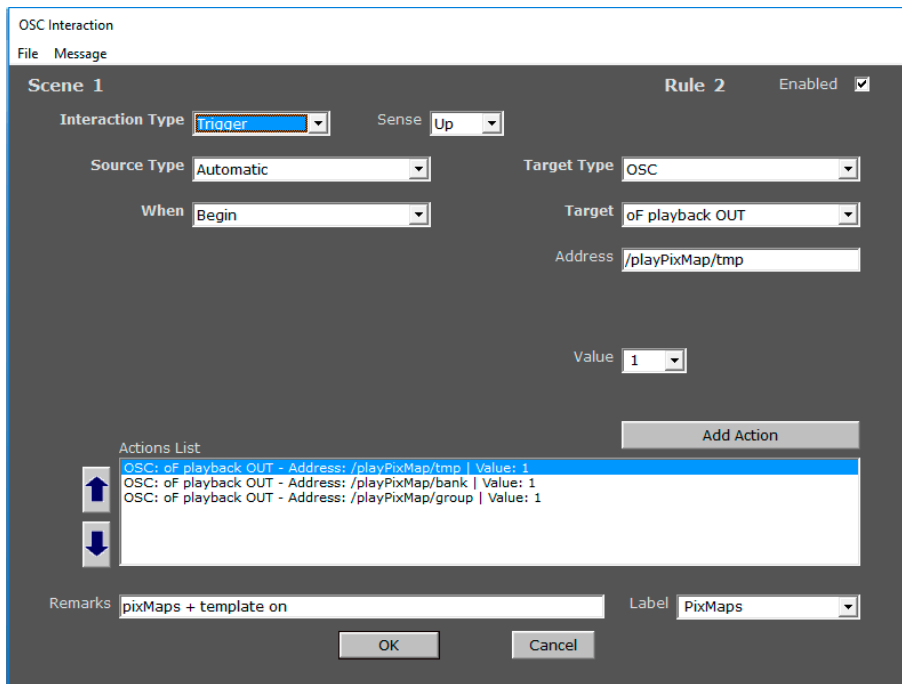
Cargar los mensajes que permiten ejecutar PIXMAPS, un group de templates, una template y un video.

Al cambiar de parte cargamos automáticamente los elementos que necesita PIXMAPS para su perfecta ejecución. La carga (begin) y la descarga (end) de estos elementos son imprescindibles si queremos utilizar PIXMAPS, sin ellos no funciona.

La rule tiene 3 events. El target **OSC**, envía por el canal **oF playback OUT**, tres address :
/playPixMap/tmp para la template value nº 1. Con este mensaje cargamos la template 1 de la lista que contiene el grupo de templates [Group] que también cargamos en esta rule.

/playPixMap/Bank para el banco de templates value nº 1. Con este mensaje cargamos el VÍDEOBANK nº1 y con él primer vídeo que contenga.

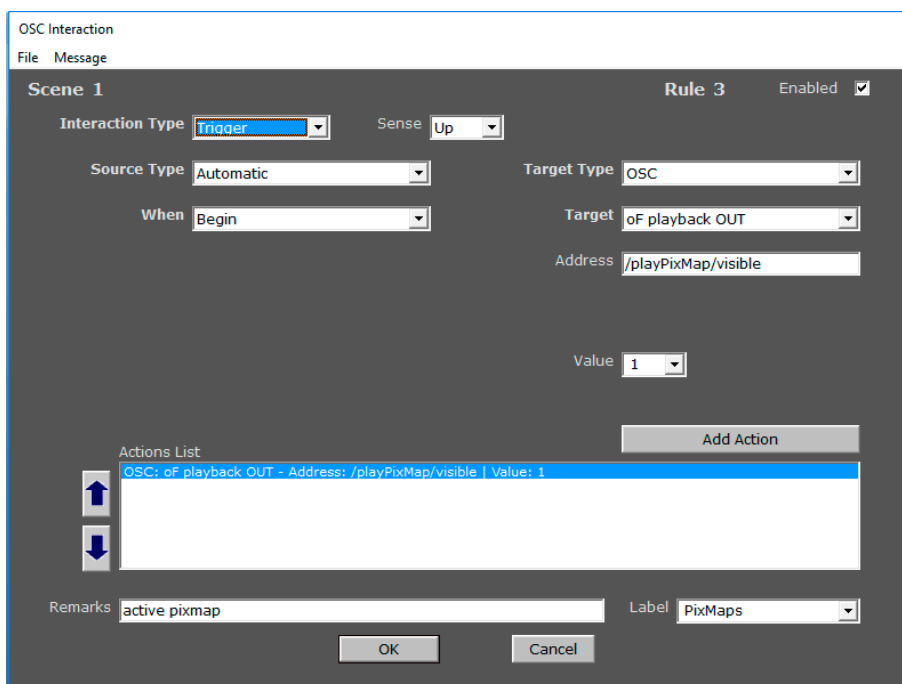
/playPixMap/Group para el grupo de Templates value nº1. Con mensaje cargamos el grupo de templates nº1.



03 rule 001 SCENE.

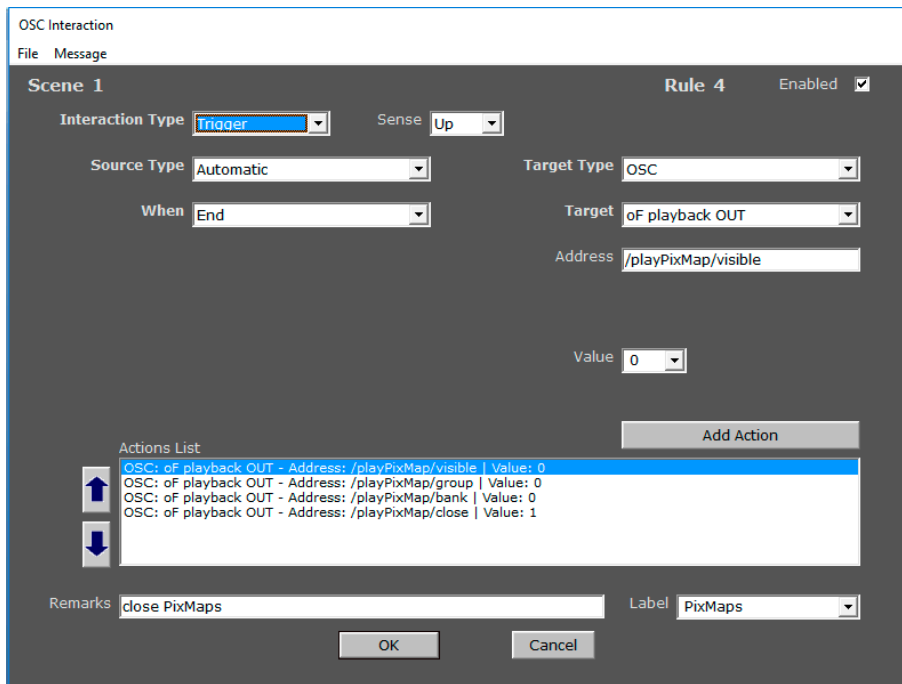
Hacemos visible automáticamente con la carga de la scene PIXMAPS.

El target type **OSC**, envía por el canal **oF playback OUT**, la address **/playPixMap/Visible** con la value **1**. Con 0 la hacemos invisible.



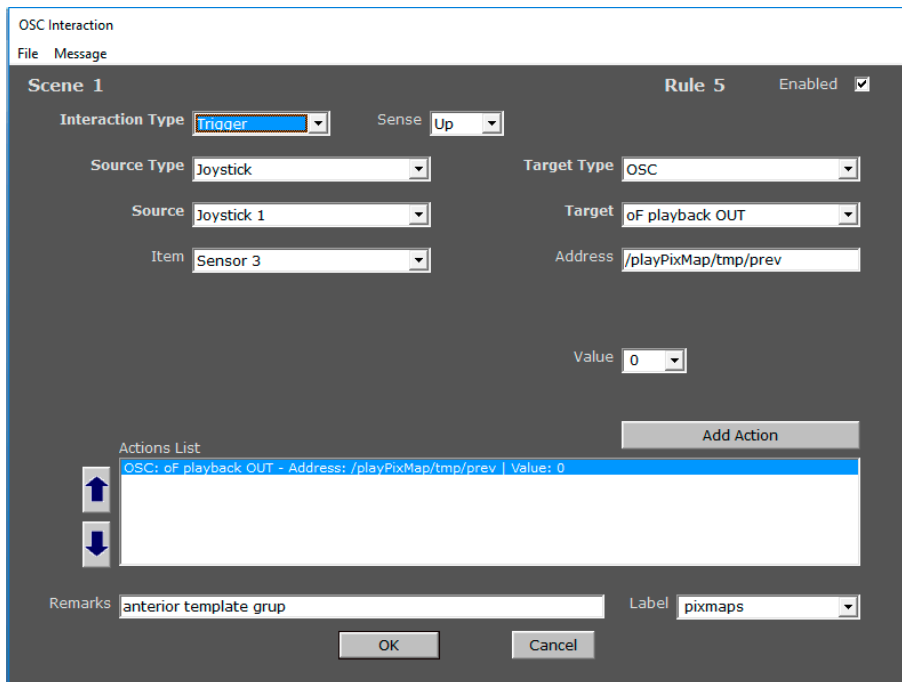
04 rule 001 SCENE

Automáticamente descargamos Group, Bank, Template y cerramos PIXMAP al cerrar la scene.



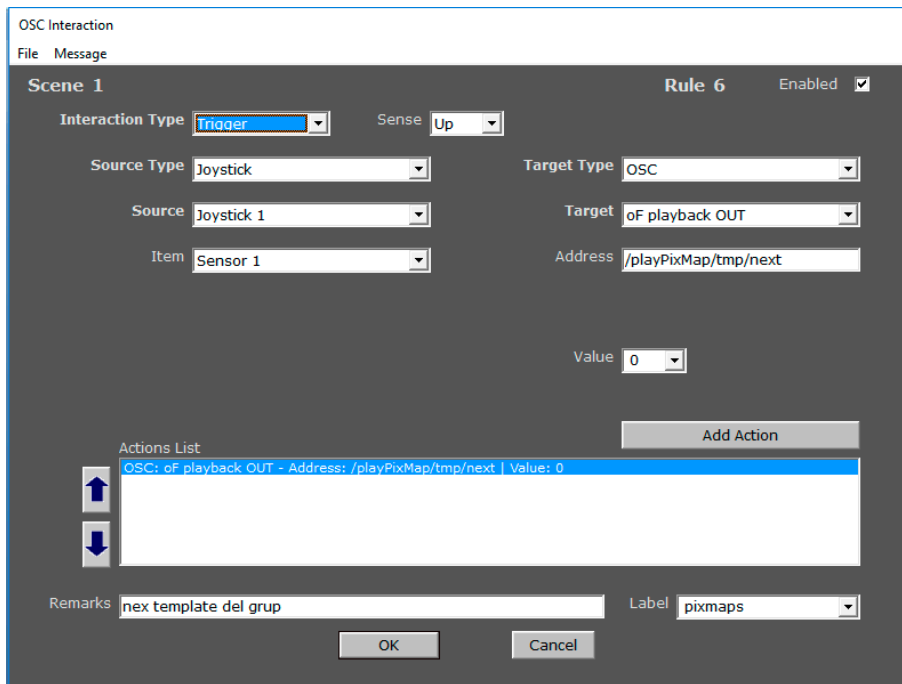
05 rule 001 SCENE.

Con el sensor 3 del joystick 1 activamos el mensaje, address, **/PLayPixMap/temp/prev** que nos permite cargar la template previa del group que hemos elegido.



06 rule 001 SCENE.

Con el sensor 1 del joystick cargamos el mensaje, address, **/PLayPixMap/temp/next** próxima template del group que hemos elegido.



002 BANK_VIDEO+PIXMAPS SCENE.

Nos movemos hacia adelante y hacia atrás en todos los VIDEOBANKS y todas las templates.

Scenes	Sources	#	Source	Target	Type	Label	Remarks	Description
000 CAMERA	<input checked="" type="checkbox"/> AUT	1	AUTO	OSC	TRIG	pixmaps	activa pixmaps+grup1+temlat...	Automatic -> OSC Trigger: Up Begin
001 VIDEO_PIXMAPS	<input checked="" type="checkbox"/> POL	2	AUTO	OSC	TRIG	pixmaps	DESACTIVA_pixmaps+grup2+...	Automatic -> OSC Trigger: Up End 0
002 BANK_VIDEO+PIXMAPS	<input checked="" type="checkbox"/> TOT	3	TOT	OSC	TRIG	pixmaps	sequent bank videos	Joystick -> OSC Trigger: Up Joystick 1
003 DIR+TEMPLATE+GROUPS	<input checked="" type="checkbox"/> TOT	4	TOT	OSC	TRIG	pixmaps	anterior bank videos	Joystick -> OSC Trigger: Up Joystick 1
004 CAMERA ALFAVIDEObANK	<input checked="" type="checkbox"/> TOT	5	TOT	OSC	TRIG	pixmaps	anterior template grup	Joystick -> OSC Trigger: Up Joystick 1
005 BANK+TEMPLATES+SEQ_PNG	<input checked="" type="checkbox"/> FLA	6	TOT	OSC	TRIG	pixmaps	nex template del grup	Joystick -> OSC Trigger: Up Joystick 1
006 FLASH								
007 PART CHANGING	<input checked="" type="checkbox"/> DIO							
008 POL_COUNTER_VARIABLES	<input checked="" type="checkbox"/> MIDI							
009 LIGHTS CONTROL (D-LIGHT)	<input checked="" type="checkbox"/> OSC							
010 END								

Veras que esta scene tiene 6 rules, puesto que venimos de una scene, la **001 VIDEO_PIXMAPS SCENE**, en la que ya hemos cargado gracias a PDM (Pol Display Manager) la aplicación Open Frameworks en el laptop de gráficos. No es necesario cargarla de nuevo, pero si viniéramos de otra aplicación de gráficos en la anterior escena tendríamos que añadir esta rule:

[Con el target de **POL, Switch Player To App**, llamamos, gracias Pol Display Manager instalada en el laptop de gráficos, **\\pseudograf3\POL4**, la aplicación Open Frameworks. Es programa que ejecuta PIXMAPS, aquí como video App OF **pixMaps**]

01 rule 002 SCENE.

Cargamos automáticamente al inicio la jerarquía necesaria para situarnos dentro del banco de video que deseemos y desde allí activar los videos.

La source nos permite cargar los 4 evens **OSC | oF playback OUT** que tiene las siguientes address:

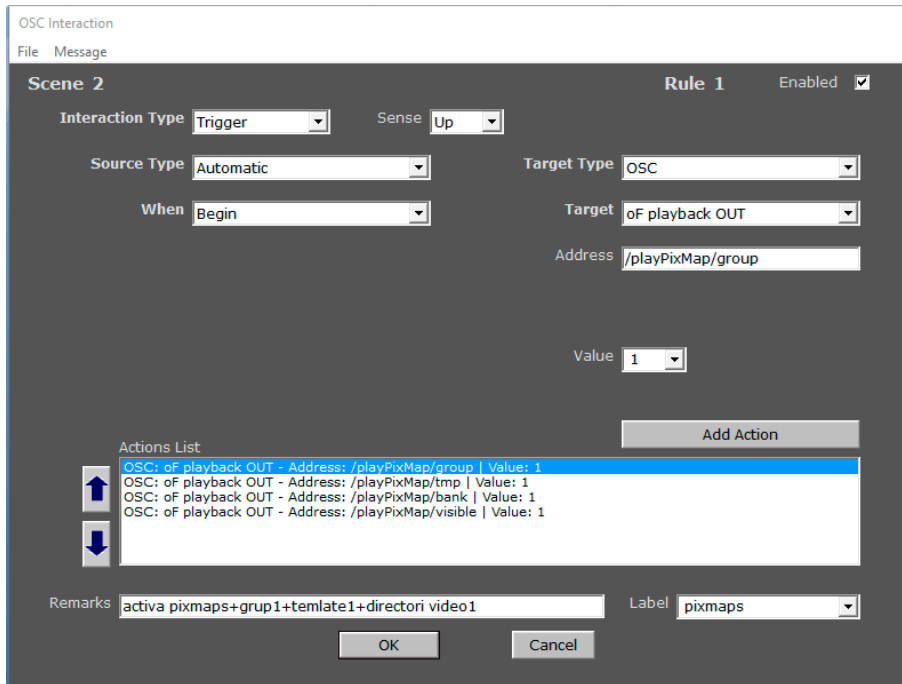
/playPixMap/group | para el grupo de templates value nº 1. Elegimos el grupo 1

/playPixMap/tmp | para el número de template del grupo actual, en nuestro caso el group 1.

La template tiene la value nº 1, en este group es la template 1 sin efectos.

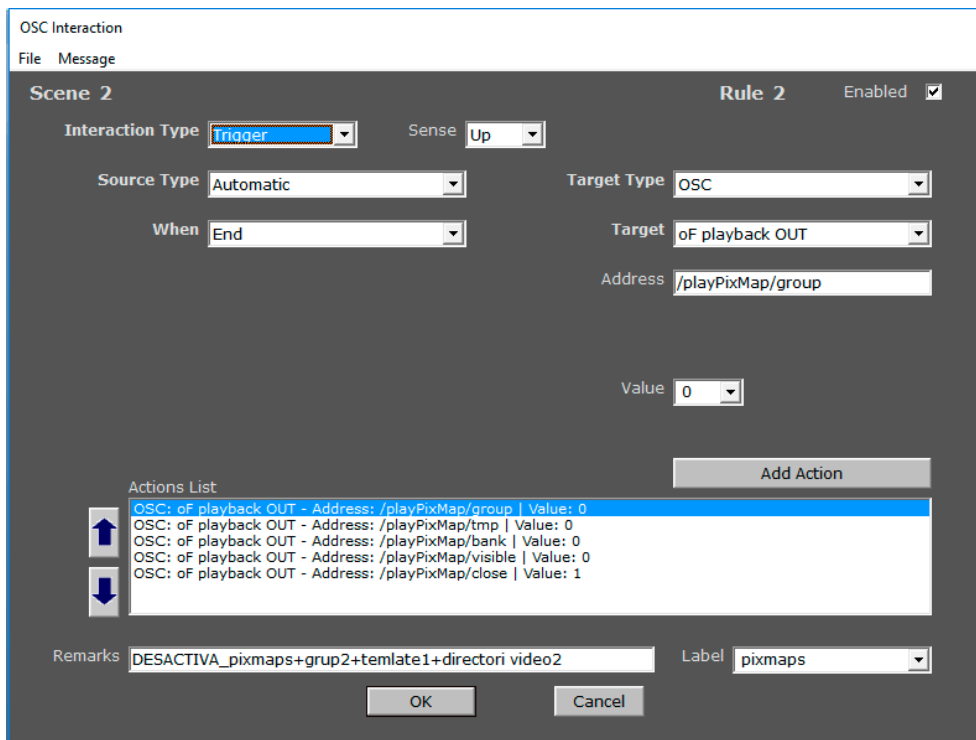
/playPixMap/bank | para el banco videos de value nº 1, este video es el que

/playPixMap/visible | carga el programa



02 rule 002 SCENE.

Descargamos automáticamente al final de la scene toda la jerarquía que hemos cargado para como veremos para situarnos dentro del un banco de video y activar los vídeos. Cerramos con esta rule las ordenes establecidas en la rule anterior pero con valor 0.

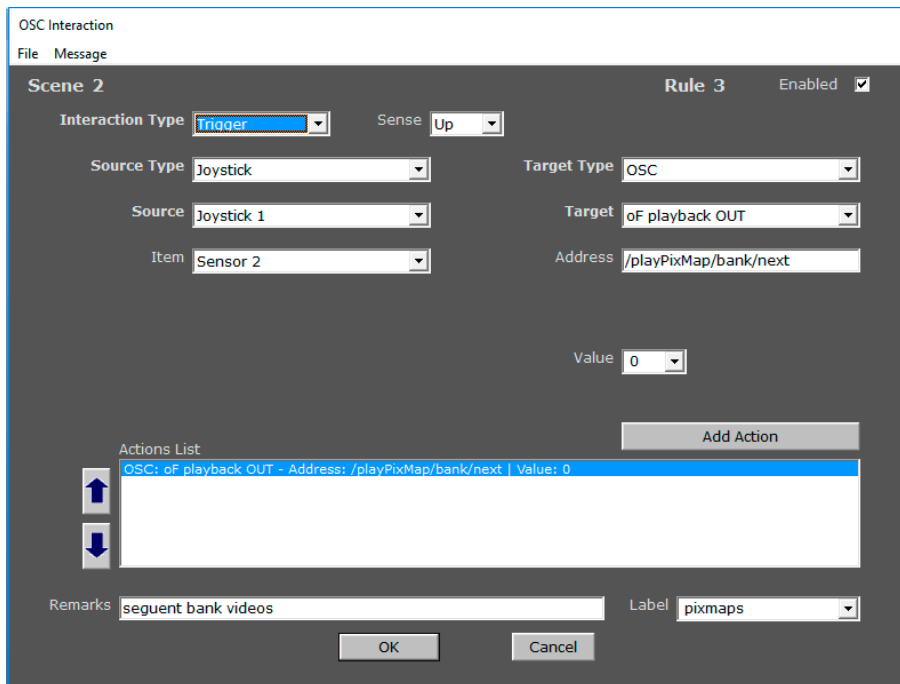


03 rule 002 SCENE.

Con el sensor 2 on/off del joystick 1 activamos el próximo vídeo.

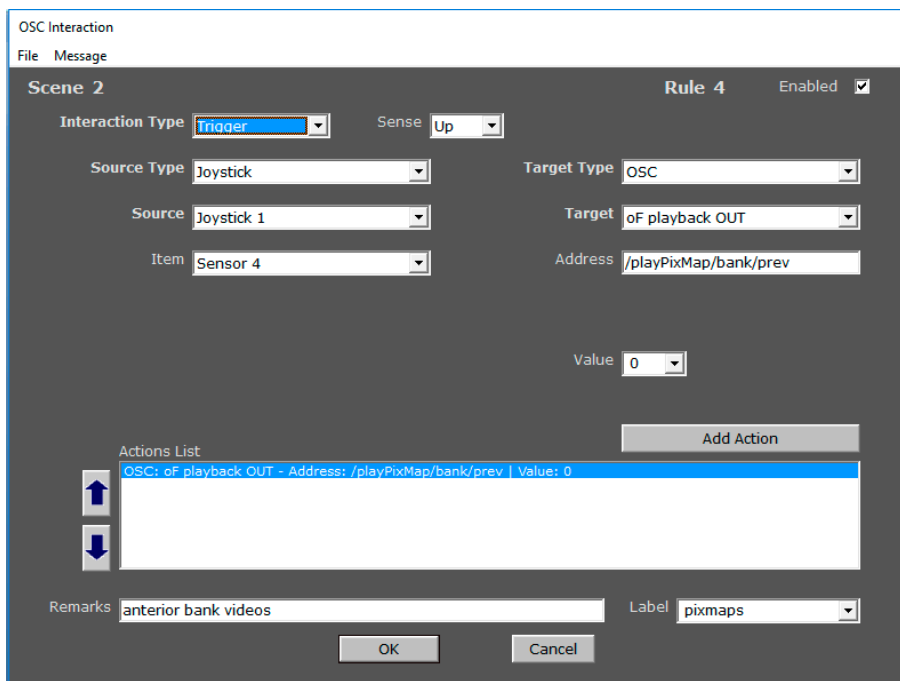
La source es Joystick | Joystick 1 | sensor 2 que activa OSC | oF playback OUT | con la address: /playPixMap/bank/next

Esta address es absoluta y por tanto no necesita de Value que permanece en 0.



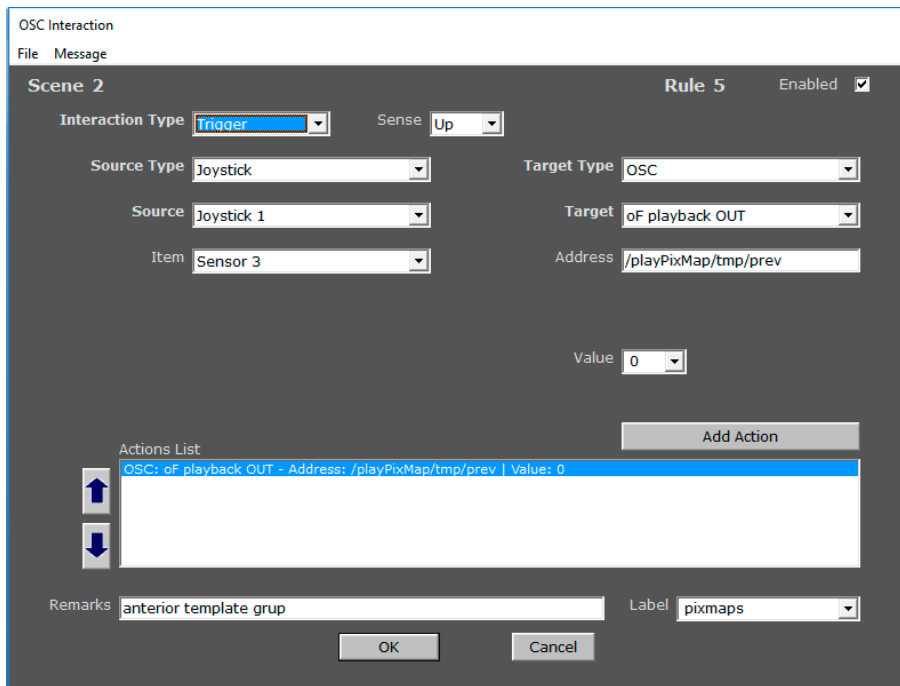
04 rule 002 SCENE.

Con el sensor 2 on/off del joystick 1 activamos el vídeo anterior del banco elegido. La source es Joystick | Joystick 1 | sensor 2 que activa OSC | oF playback OUT | con la address: **/playPixMap/bank/prev** | Esta address no necesita de Value, que permanece en 0.



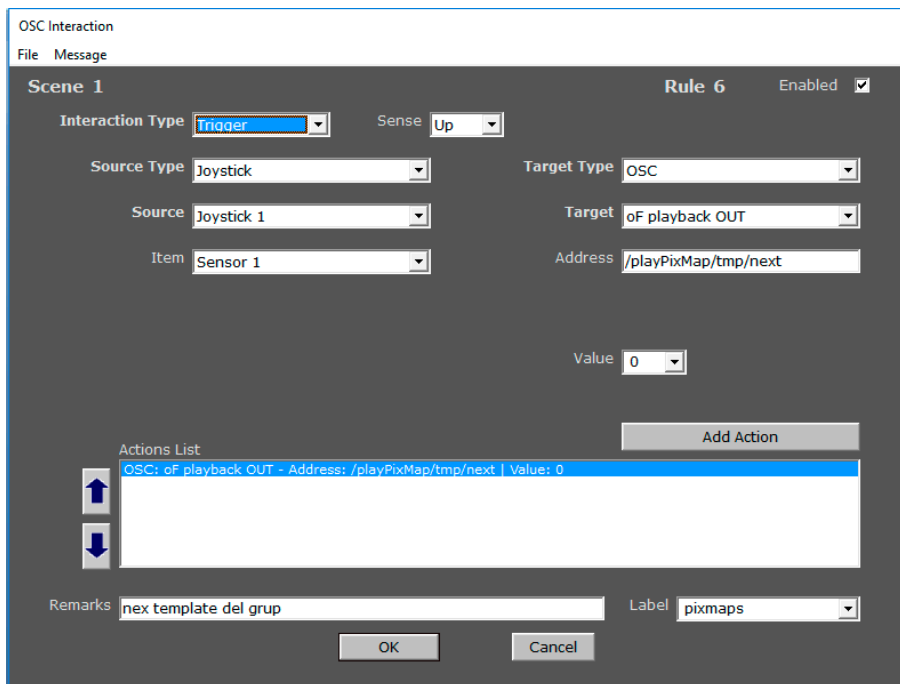
05 rule 006 SCENE.

Con el sensor 3 on/off del joystick 1 activamos la template anterior del group elegido. La source es Joystick | Joystick 1 | sensor 3 que activa OSC | oF playback OUT | con la address: **/playPixMap/temp/prev** | Esta address no necesita de Value, que permanece en 0.



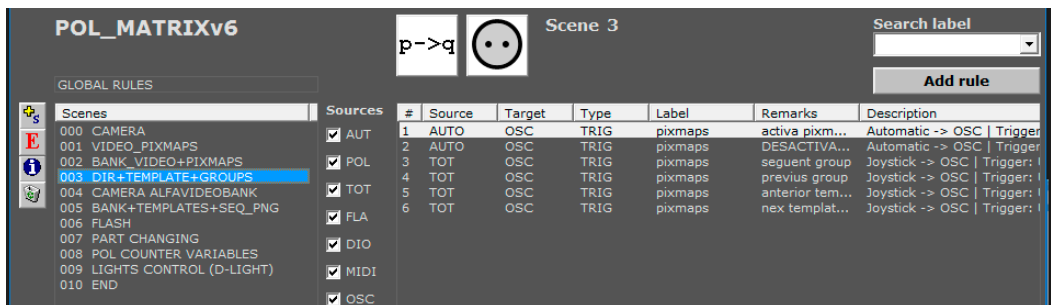
06 rule 002 SCENE.

Con el sensor 1 on/off del joystick 1 activamos la template posterior del group elegido. La source es Joystick | Joystick 1 | sensor 1 que activa OSC | oF playback OUT | con la address: **/playPixMap/temp/prev** | Esta address no necesita de Value, que permanece en 0.



003 DIR+TEMPLATE+GROUPS SCENE.

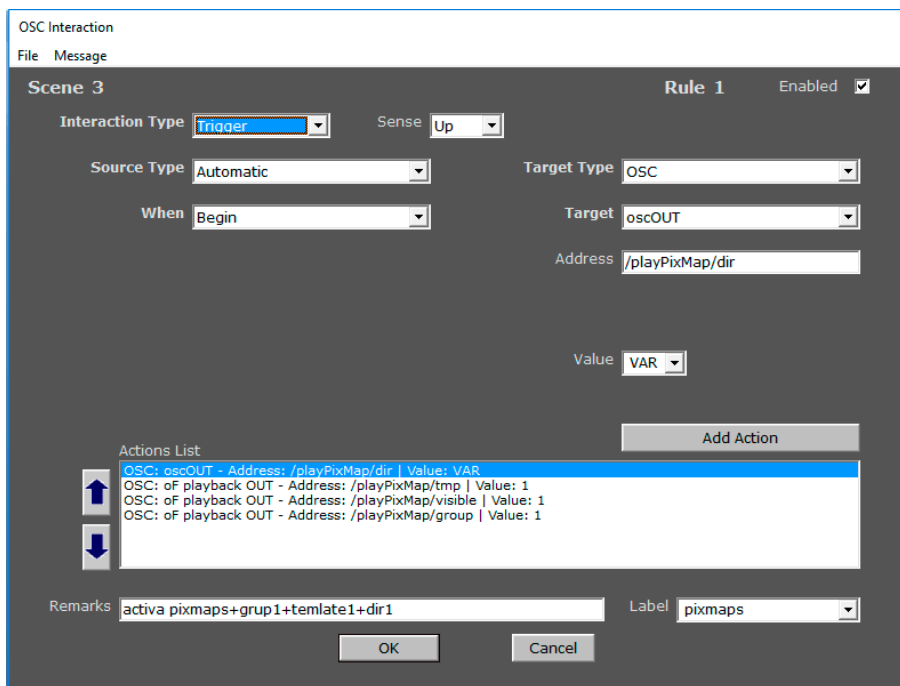
Con esta scene cargamos las capturas los bancos de las capturas vídeo y podemos movernos diferentes templates mosaico de hasta 32x24 vídeos.



Como verás, y si estás leyendo este manual del tirón este documento, algunas rules se repiten.

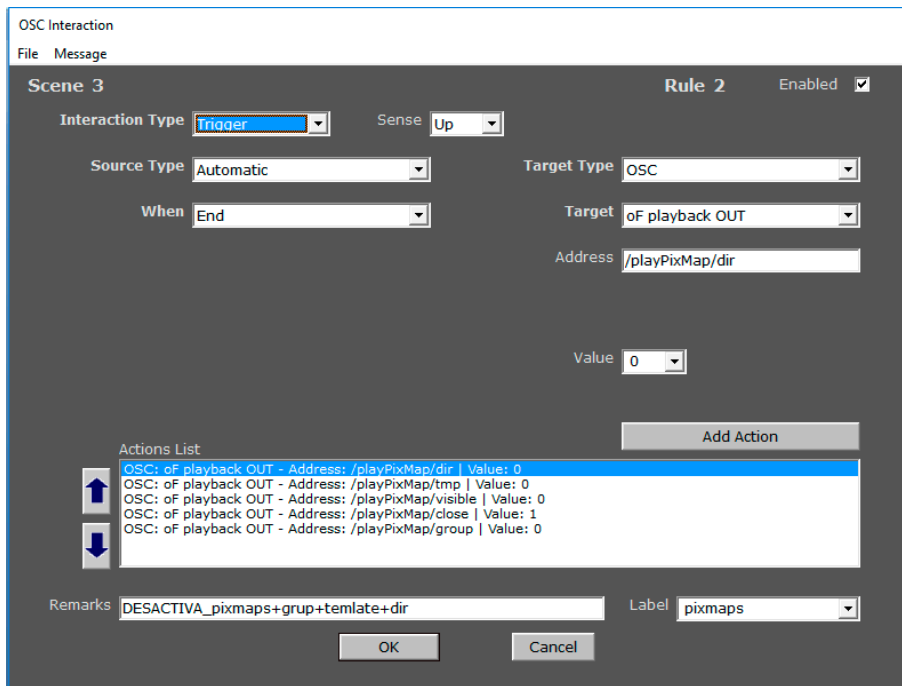
01 rule. 003 SCENE

Cargamos automáticamente al inicio en esta rule toda la jerarquía necesaria para situarnos en un entorno de PIXELMAPS. La rule es igual que **01 rule 002 SCENE**, pero cambia el mensaje de VIDEOBANK, **/playPixMap/bank**, por el de VIDEOFOLDER **/playPixMap/dir** que carga las carpetas de capturas.



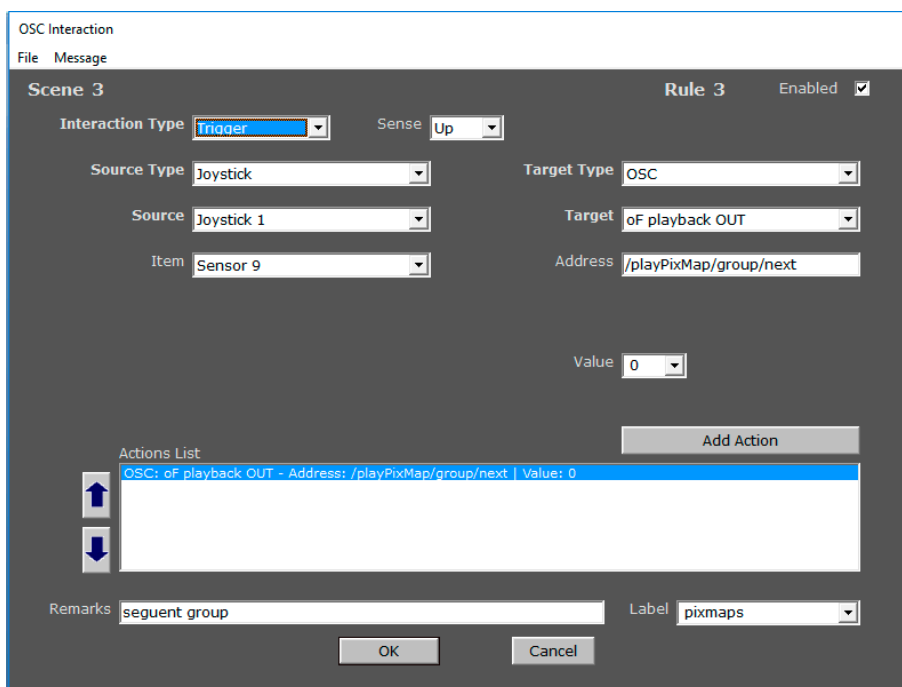
02 rule. 003 SCENE

Descargamos automáticamente al final en esta scene toda la jerarquía que hemos cargado en al rule anterior.



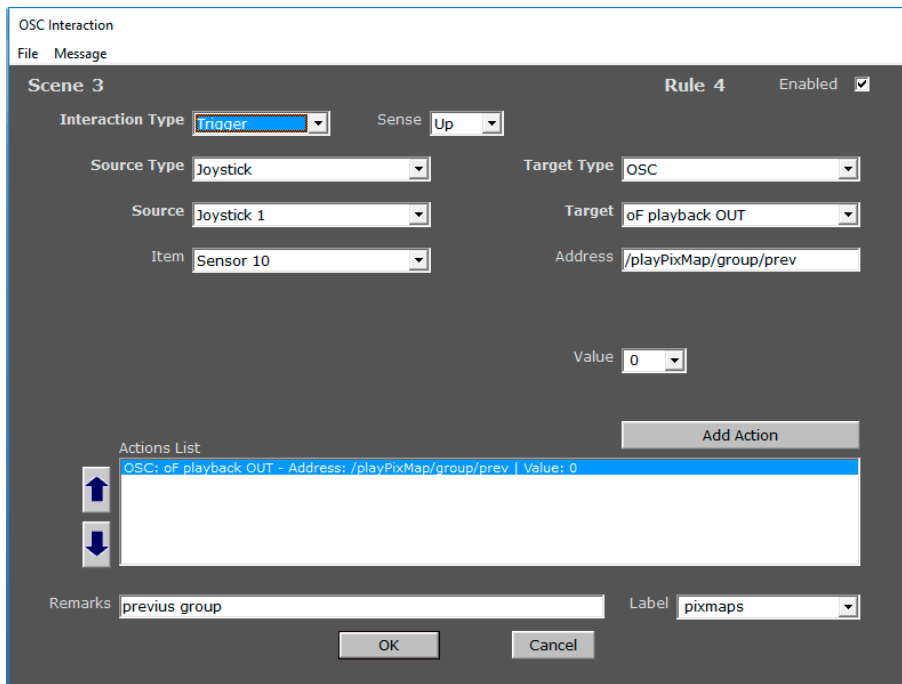
03 rule 003 SCENE.

Con el sensor 5 on/off del joystick 1 activamos el próximo group de templates.
 La source es Joystick | Joystick 1 | sensor 5 que activa OSC | oF playback OUT | con la address: **/playPixMap/group/next** | Esta address no necesita de Value, que permanece en 0.



04 rule 003 SCENE.

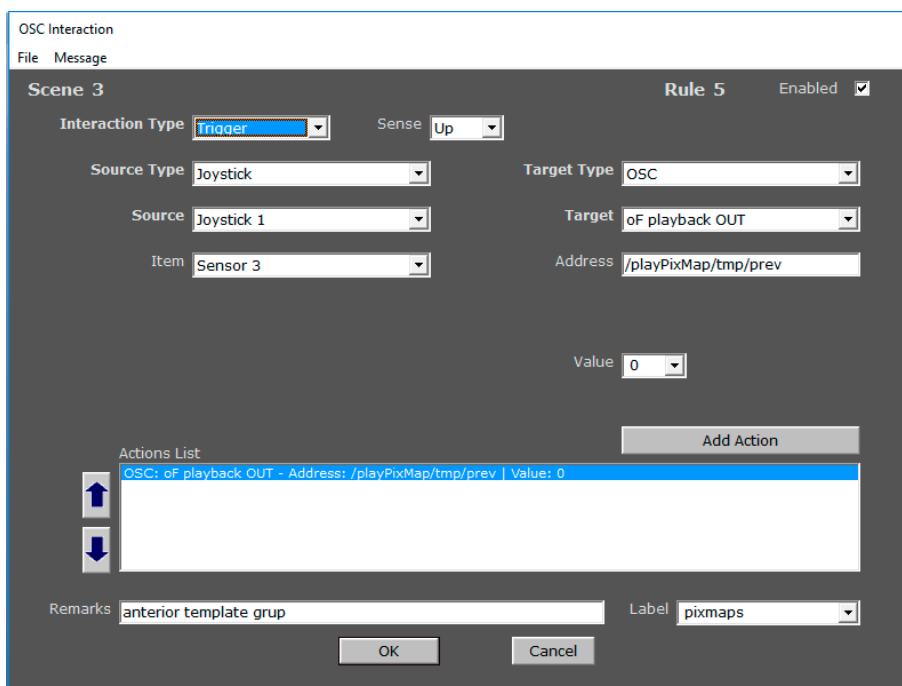
Con el sensor 7 on/off del joystick 1 activamos el group anterior de templates.
 La source es Joystick | Joystick 1 | sensor 7 que activa OSC | oF playback OUT | con la address: **/playPixMap/group/prev** | Esta address no necesita de Value, que permanece en 0.



05 rule 003 SCENE.

Con el sensor 3 on/off del joystick 1 activamos el próximo template.

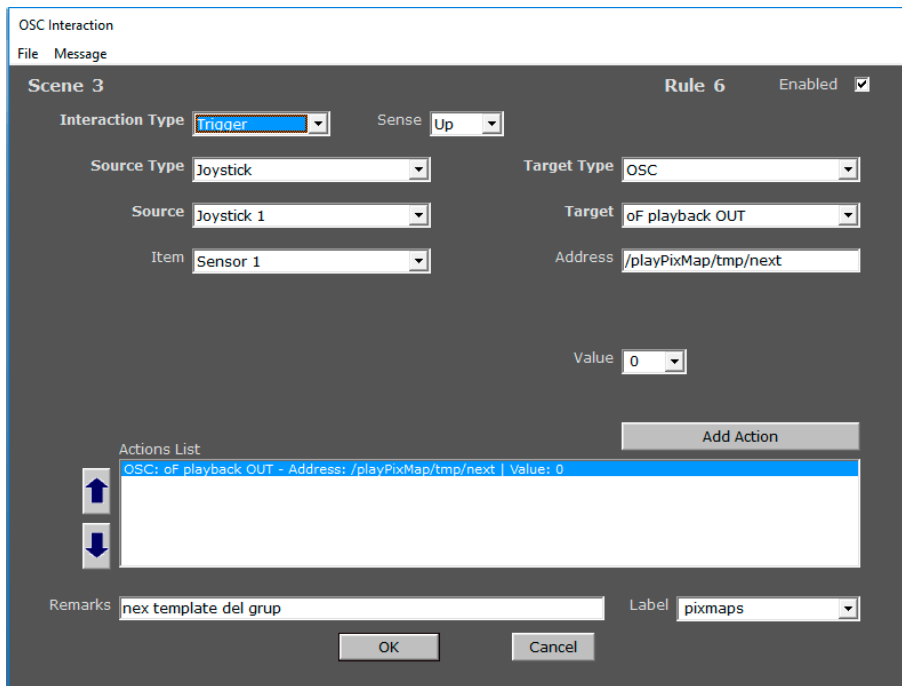
La source es Joystick | Joystick 1 | sensor 3 que activa OSC | oF playback OUT | con la address: **/playPixMap/tem/next** | Esta address no necesita de Value, que permanece en 0.



06 rule 003 SCENE.

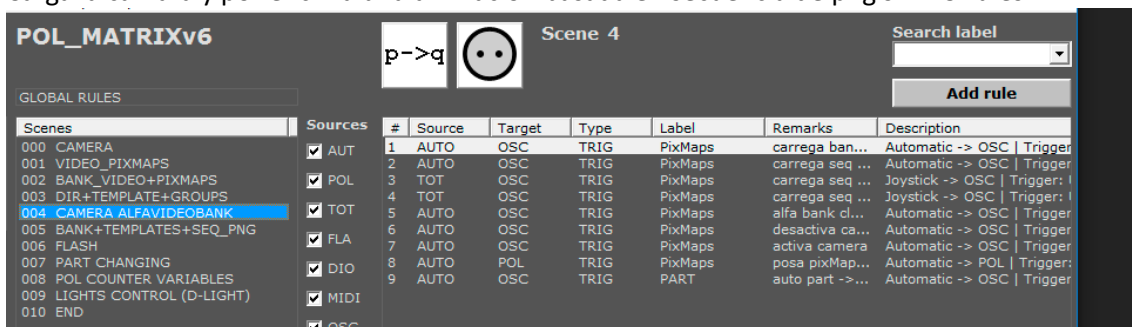
Con el sensor 3 on/off del joystick 1 activamos el próximo template.

La source es Joystick | Joystick 1 | sensor 3 que activa OSC | oF playback OUT | con la address: **/playPixMap/tem/next** | Esta address no necesita de Value, que permanece en 0.



004 CAMERA+ALFAVIDEOBANK SCENE

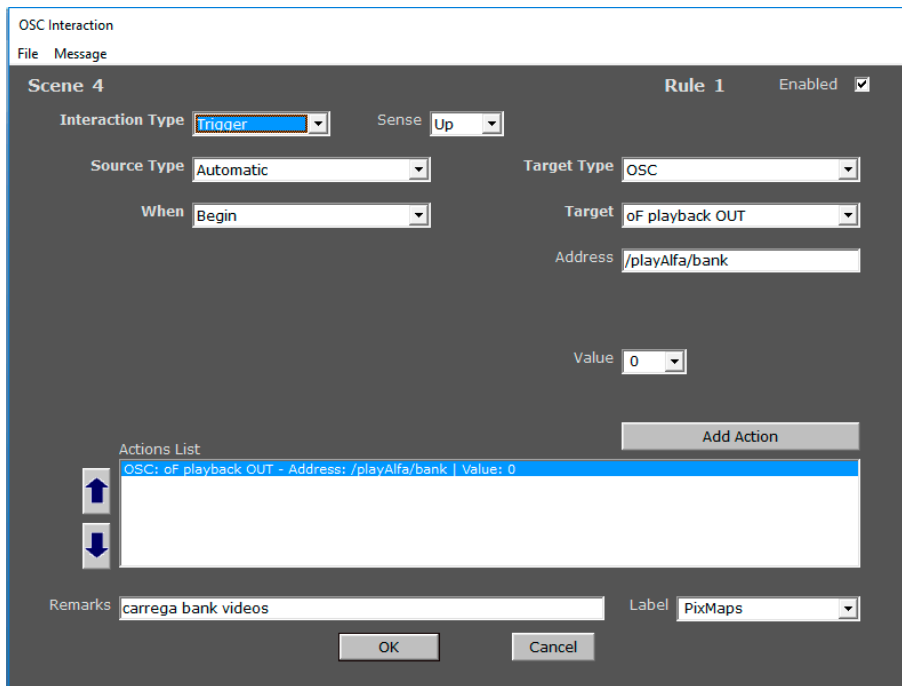
Carga la cámara y por encima una animación basada en secuencia de png's. En 9 rules.



01 rule 004 SCENE.

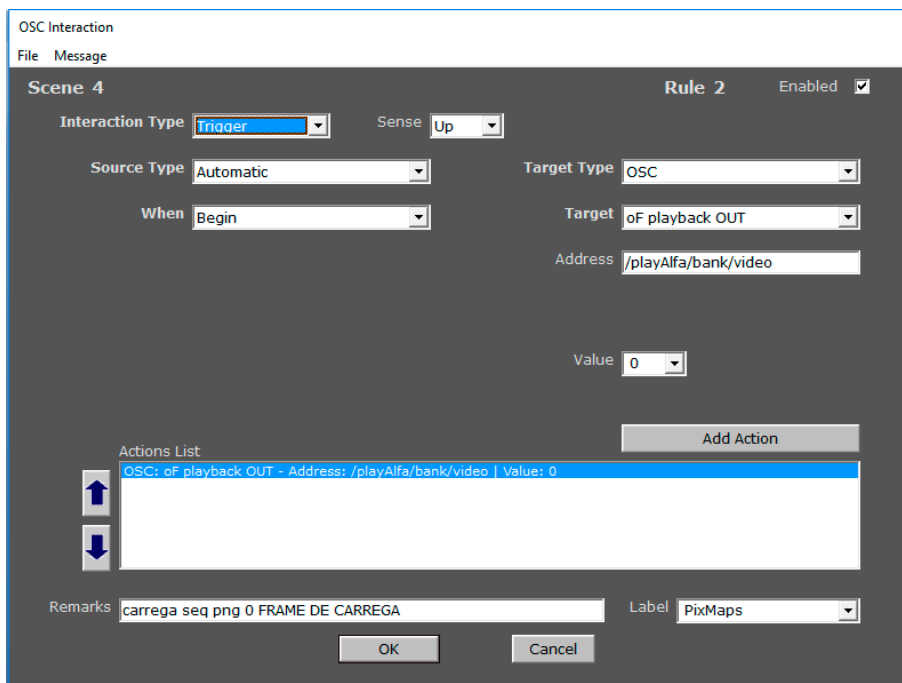
Automaticamente carga el entorno de alfas, sin este mensaje no puedes cargar los grupos de png's.

La source es Automatic |Begin | que activa OSC | oF playback OUT | con la address: /playAlfa/bank | Esta address necesita de Value, que es siempre 0.



02 rule 004 SCENE.

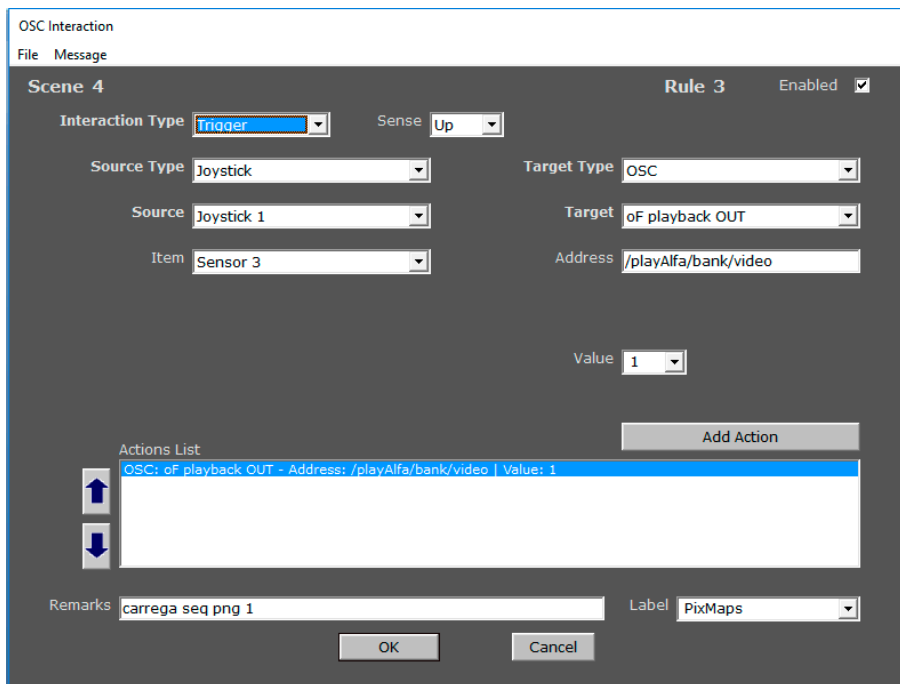
Carga una carpeta con png's en alfa y se reproducen sobre la base que hemos configurado. La source es Automatic | Begin | que activa OSC | oF playback OUT | con la address: **/playAlfa/bank/video** | Esta address necesita de Value y su numeración nos da la posición en la lista de carpetas de ALFAVIDEOBANKS. En esta versión de PIXMAPS cuenta desde 0,1,2... el valor 0 será la primera carpeta.



03 rule 004 SCENE.

Desde el sensor 3 del Joystick 1 carga el alfavideobank segundo. La source es Joystick | Joystick 1 | sensor 3 | que activa OSC | oF playback OUT | con la

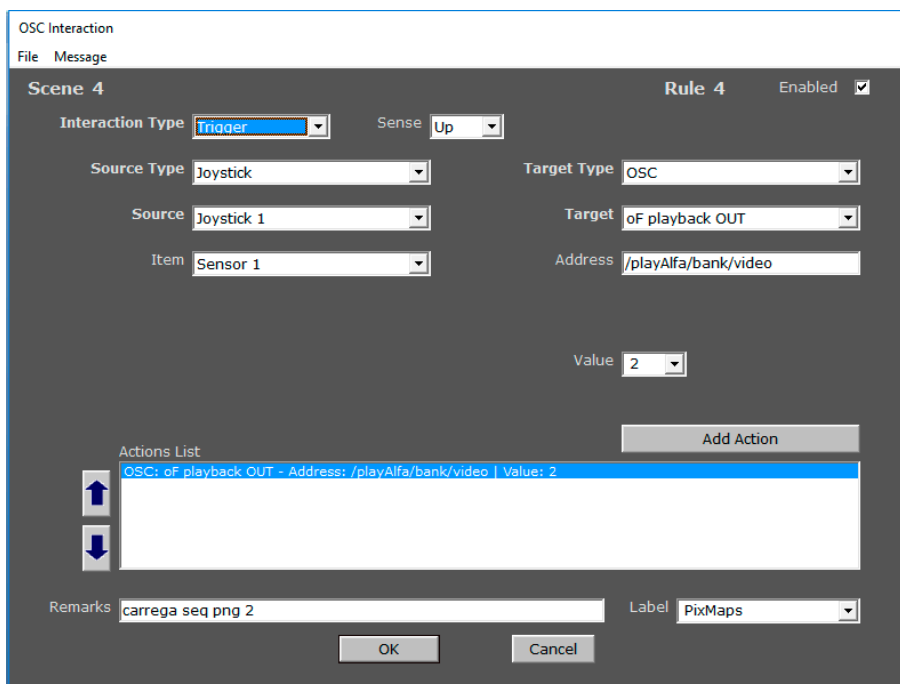
address: **/playAlfa/bank/video** | La Value nos da la posición en la lista de carpetas de ALFAVIDEOBANKS. En esta versión de PIXMAPS cuenta desde 0,1,2... el valor 1 será la segunda carpeta.



04 rule 004 SCENE.

Desde el sensor 1 del Joystick 1 carga el alfavideobank tercero.

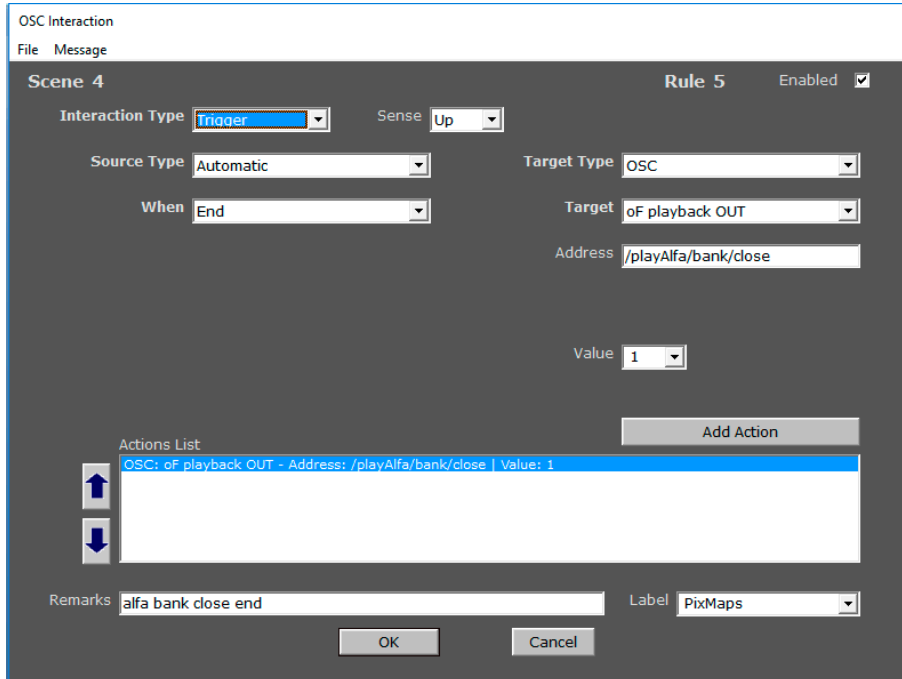
La source es Joystick | Joystick 1 | sensor 1 | que activa OSC | oF playback OUT | con la address: **/playAlfa/bank/video** | La Value nos da la posición en la lista de carpetas de ALFAVIDEOBANKS. En esta versión de PIXMAPS cuenta desde 0,1,2... el valor 2 será la tercera carpeta.



05rule 004 SCENE.

Automáticamente descarga el entrono de alfas, al cambiar de parte.

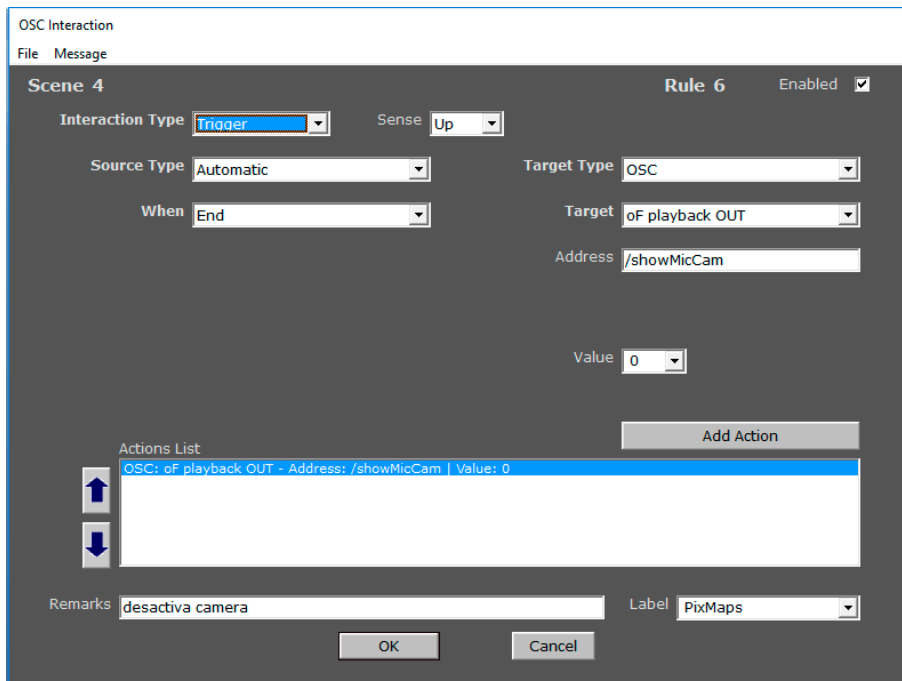
La source es Automatic |End | que activa OSC | oF playback OUT | con la address: /playAlfa/bank/close | Esta address necesita de Value, que es siempre 1.



06 rule 004 SCENE.

Desactiva la cámara al salir de la scene.

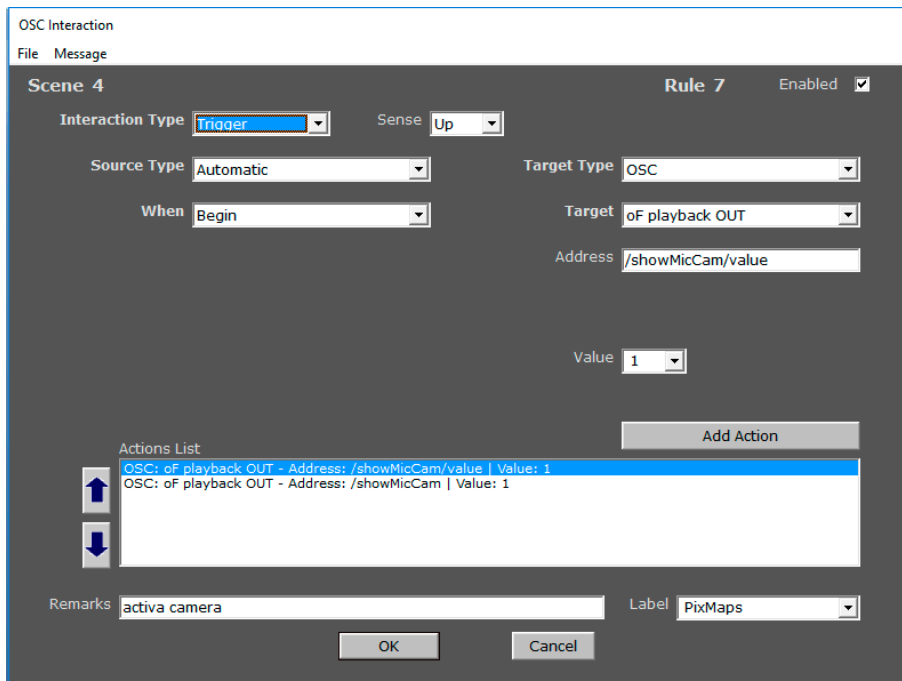
La source es Automatic |End | que activa OSC | oF playback OUT | con la address: /showMicCam | Esta address necesita de Value, que es siempre 0.



07 rule 004 SCENE.

Activa la cámara al entrar en la scene.

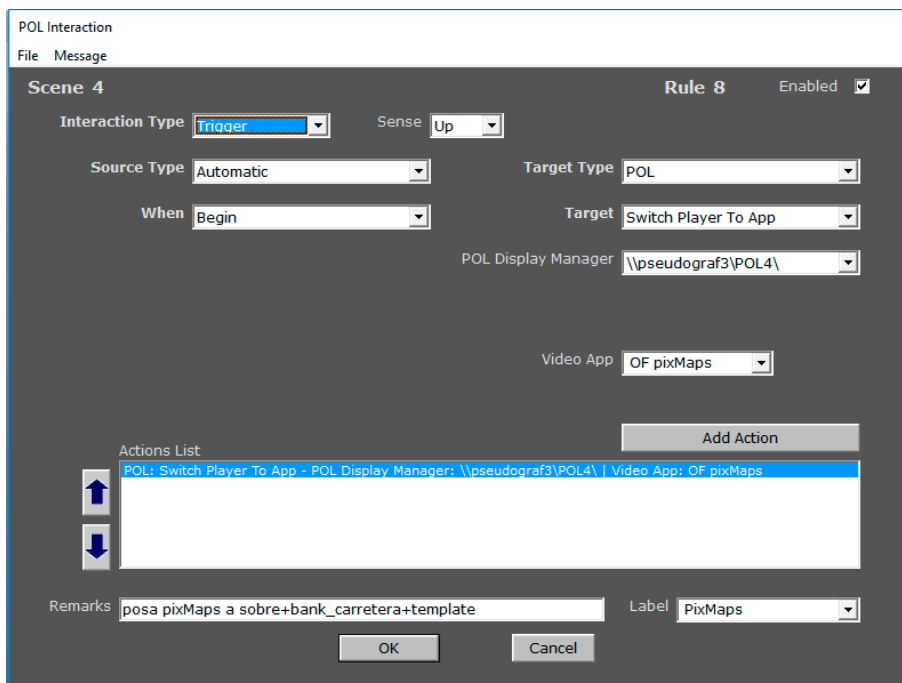
La source es Automatic |Begin | que activa OSC | oF playback OUT | con la address: /showMicCam /value|Esta address necesita de Value 1. Segundo even de esta rule con la misma ruta pero com la address: /showMicCam |Esta address necesita de Value 1.



08 rule 004 SCENE.

Llevar una aplicación de gráficos con PDM al primer término de la pantalla.

La source es Automatic |Begin | que activa POL | Switch Player To App | con la address: \\pseudograf3\POL4 | OF pixMap | nos lleva al frente la aplicación PIXMAPS



005 BANK+TEMPLATES+SEQ_PNG SCENE.

Carga distintas animaciones en png's y por debajo todo el sistema de templates y videos. Puedes cambiar las animaciones y los videos y los efectos de los videos.

GLOBAL RULES								Add rule
Scenes	Sources	#	Source	Target	Type	Label	Remarks	Description
000 CAMERA	<input type="checkbox"/> AUT	1	AUTO	OSC	TRIG	pixmaps	activa pixmaps+grup1+template1+directori...	Automatic -> OSC Trigger: Up Begin OSC
001 VIDEO_PIXMAPS	<input checked="" type="checkbox"/> POL	2	TOT	OSC	TRIG	pixmaps	DESACTIVA pixmaps+grup2+template1+di...	Automatic -> OSC Trigger: Up End OSC
002 BANK_VIDEO+PIXMAPS	<input checked="" type="checkbox"/> TOT	3	TOT	OSC	TRIG	pixmaps	seguent bank videos	Joystick -> OSC Trigger: Up Joystick 1 It
003 DIR+TEMPLATES+GROUPS	<input checked="" type="checkbox"/> TOT	4	TOT	OSC	TRIG	pixmaps	anterior bank videos	Joystick -> OSC Trigger: Up Joystick 1 It
004 CAMERA ALFAVIDEODANK	<input checked="" type="checkbox"/> TOT	5	TOT	OSC	TRIG	pixmaps	anterior template grup	Joystick -> OSC Trigger: Up Joystick 1 It
005 BANK+TEMPLATES+SEQ_PNG	<input checked="" type="checkbox"/> FLA	6	TOT	OSC	TRIG	pixmaps	next template del grup	Joystick -> OSC Trigger: Up Joystick 1 It
006 FLASH	<input checked="" type="checkbox"/> DIO	7	AUTO	OSC	TRIG	PixMaps	carrega alfa seq 0 FRAME DE CARREGA	Automatic -> OSC Trigger: Up Begin OSC
007 PART CHANGING	<input checked="" type="checkbox"/> MIDI	8	TOT	OSC	TRIG	PixMaps	carrega seq png 1	Joystick -> OSC Trigger: Up Joystick 1 It
008 POL_COUNTER_VARIABLES	<input checked="" type="checkbox"/> OSC	9	TOT	OSC	TRIG	PixMaps	carrega seq png 3	Joystick -> OSC Trigger: Up Joystick 1 It
009 LIGHTS CONTROL (D-LIGHT)		10	AUTO	OSC	TRIG	PixMaps		Automatic -> OSC Trigger: Up End OSC
010 END								

01 rule 005 SCENE.

Carga Alfavideobank, Groups, Templates y videobanks en una sola rule.

La source es Automatic |Begin | que activa OSC | con cinco la address:

playPixMap/group Carga el group 1 de las templates, value 1.

playPixMap/tmp Carga la template 1 de group, value 1.

playPixMap/bank Carga el videobank 1, value 1.

playPixMap/visible Hace visible Pixmaps, value 1.

playAlfa/bank Hace visible los las secuencias Alfa de png's, value 0.

OSC Interaction

File Message

Scene 5 Rule 1 Enabled

Interaction Type **Trigger** Sense **Up**

Source Type **Automatic** Target Type **OSC**

When **Begin** Target **oF playback OUT**

Address **/playPixMap/group**

Value **1**

Add Action

Actions List

- OSC: oF playback OUT - Address: /playPixMap/group | Value: 1
- OSC: oF playback OUT - Address: /playPixMap/tmp | Value: 1
- OSC: oF playback OUT - Address: /playPixMap/bank | Value: 1
- OSC: oF playback OUT - Address: /playPixMap/visible | Value: 1
- OSC: oF playback OUT - Address: /playAlfa/bank | Value: 0

Remarks **activa pixmaps+grup1+template1+directori video1** Label **pixmaps**

OK Cancel

02 rule 005 SCENE.

Descarga Alfavideobank, Groups, Templates y videobanks en una sola rule.

La source es Automatic |End | que activa OSC | con cinco la address:

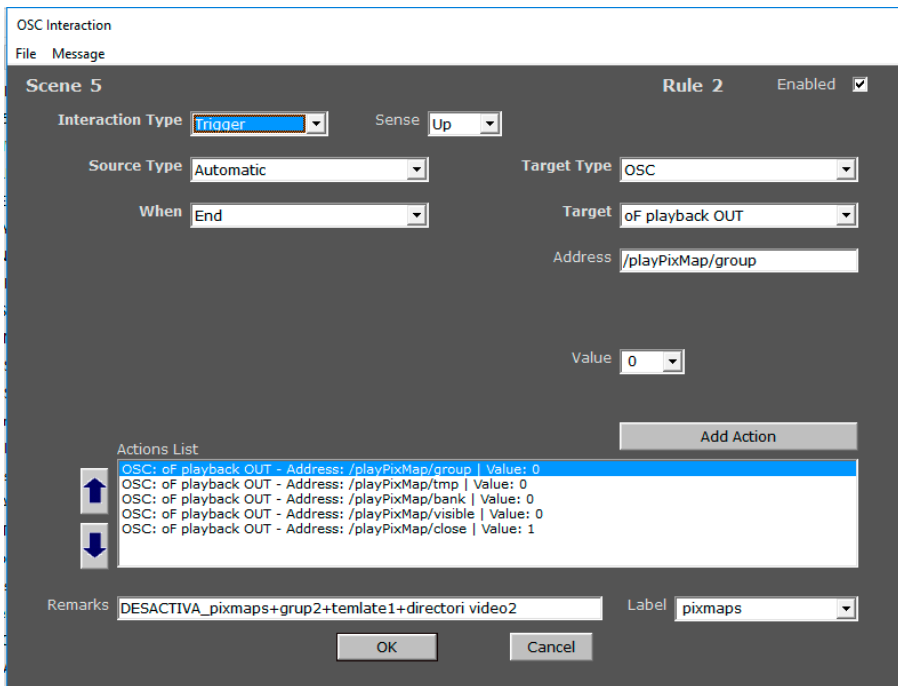
playPixMap/group descarga el group de las templates, value 0.

playPixMap/tmp descarga la template 1 de group, value 0.

playPixMap/bank descarga el videobank 1, value 1.

playPixMap/visible Hace invisible Pixmaps, value 0.

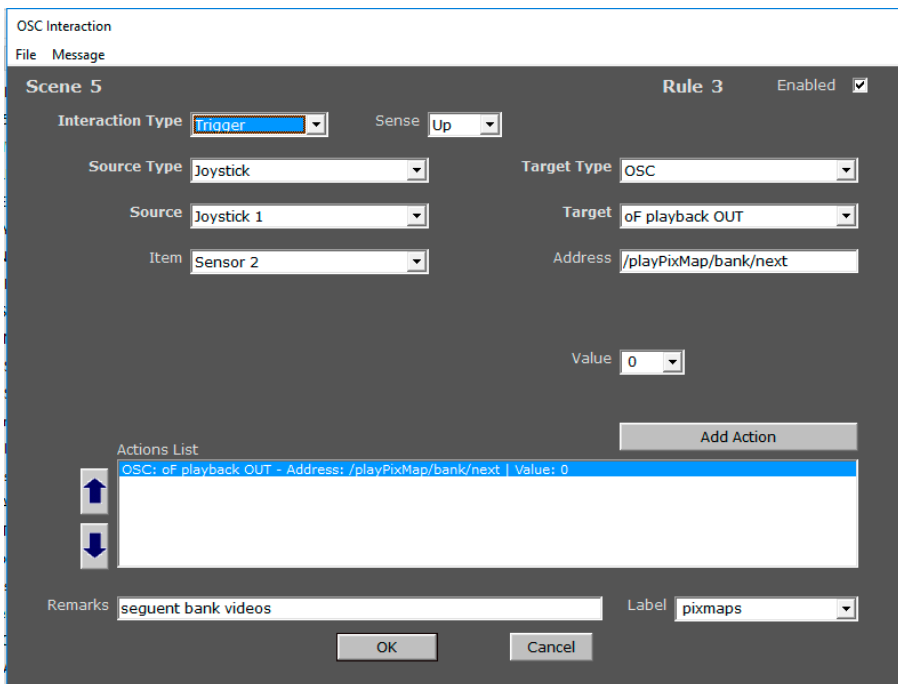
playPixMap/close Cierra el pixmaps, value 0.



03 rule 005 SCENE.

Con el sensor 2 del Joystick 1 carga el siguiente banco de videos de la lista VIDEOBANKS

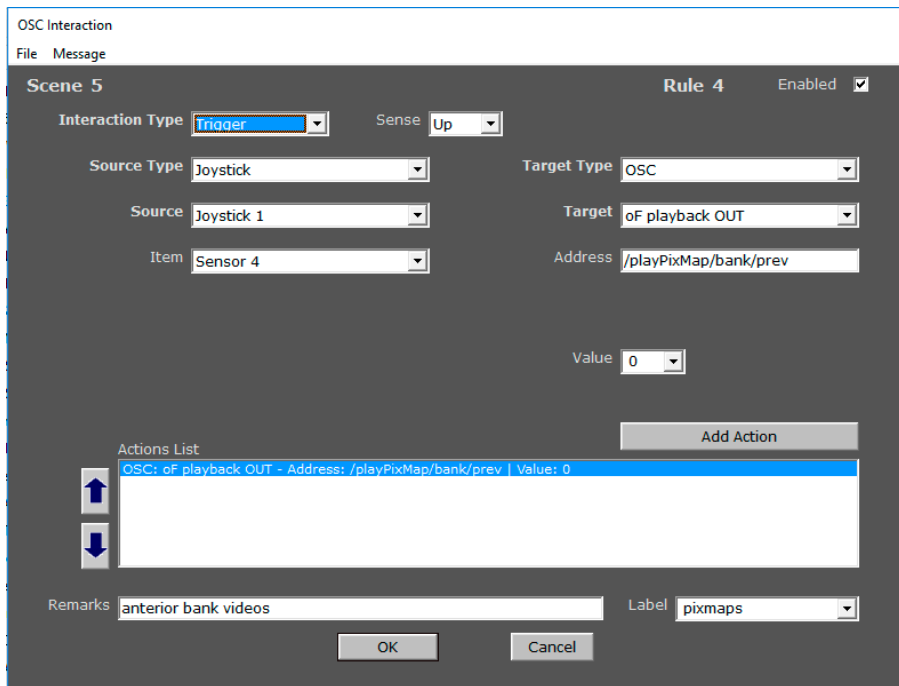
La source es Joystick | Joystick 1 | sensor 2 | que activa OSC | oF playback OUT | con la address: **playPixMap/bank/next**



04 rule 005 SCENE.

Con el sensor 4 del Joystick 1 carga el anterior banco de videos de la lista VIDEOBANKS

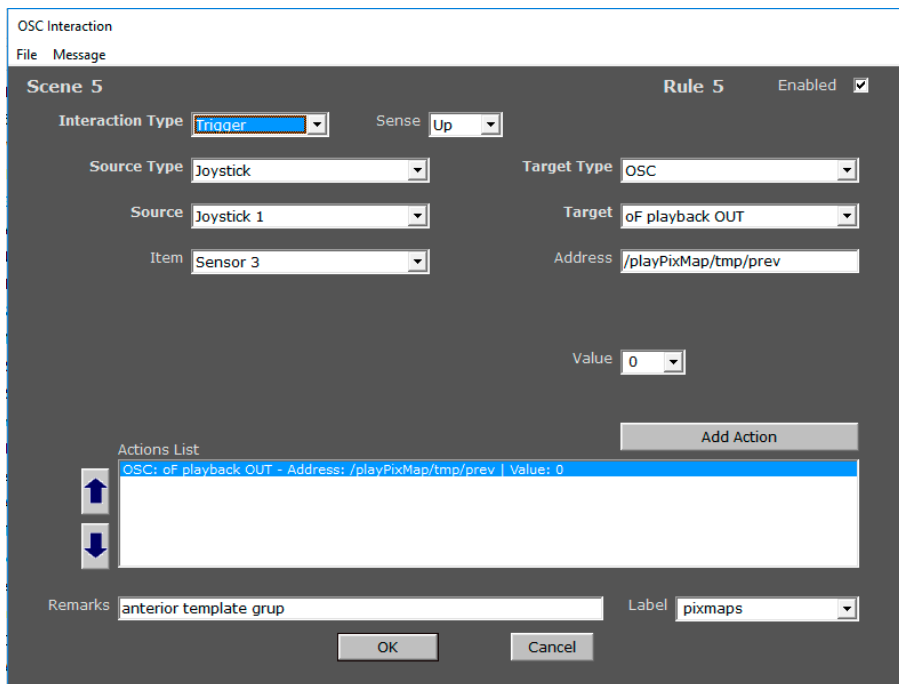
La source es Joystick | Joystick 1 | sensor 4 | que activa OSC | oF playback OUT | con la address: **playPixMap/bank/prev**



05 rule 005 SCENE.

Con el sensor 3 del Joystick 1 carga el anterior template de efectos de la lista PIXGROUP

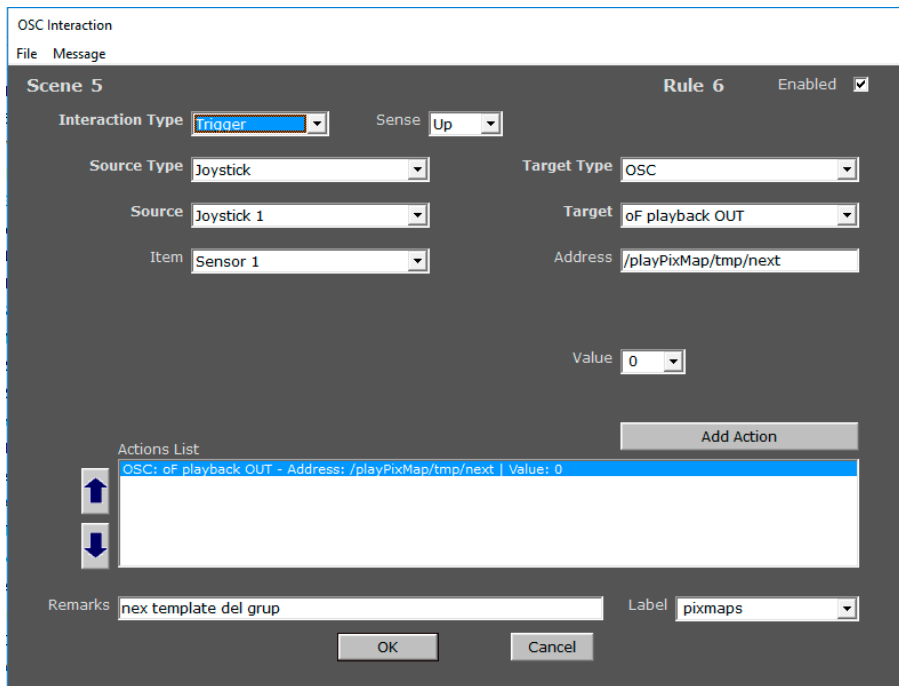
La source es Joystick | Joystick 1 | sensor 3 | que activa OSC | oF playback OUT | con la address: **playPixMap/temp/prev**



06 rule 005 SCENE.

Con el sensor 1 del Joystick 1 carga la siguiente template de efectos de la lista PIXGROUP

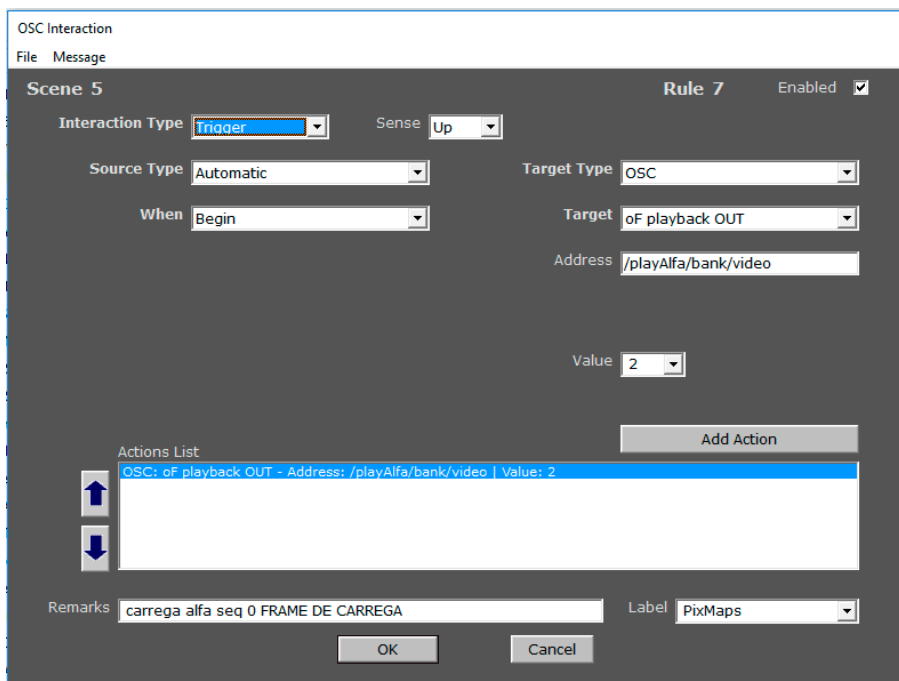
La source es Joystick | Joystick 1 | sensor 1 | que activa OSC | oF playback OUT | con la address: **playPixMap/temp/next**



07 rule 005 SCENE.

Carga una animación al entrar en la scene.

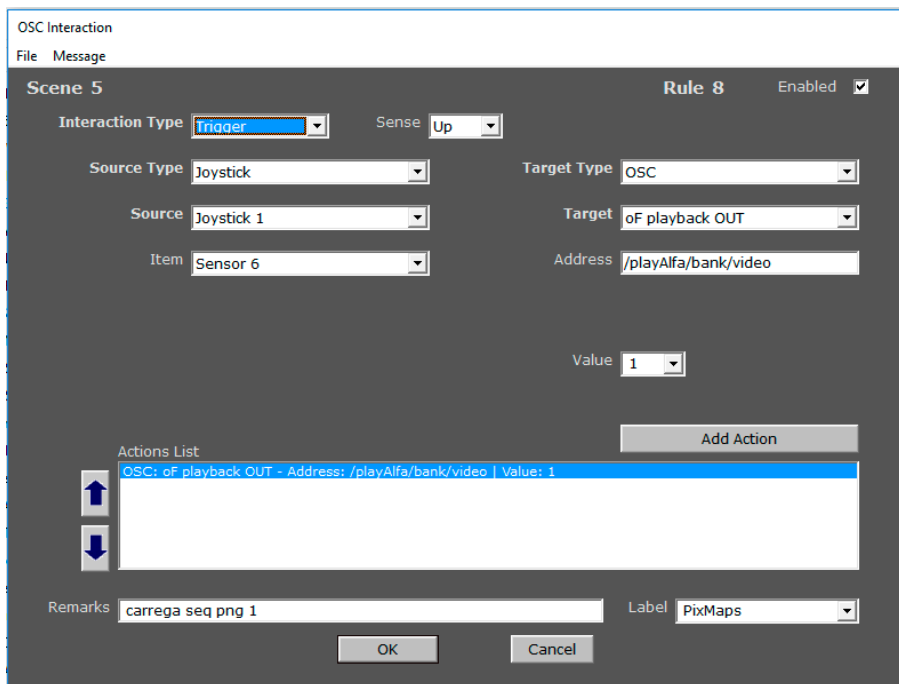
La source es Automatic | Begin | que activa OSC | con cinco la address: **playAlfa/bank/video /** El value es 2, es decir cargará la tercera carpeta de la lista de ALFAVIDEOBANS. Esta versión de PIXMAPS cuenta desde 0.



08 rule 005 SCENE.

Con el sensor 6 del Joystick 1 carga una secuencia específica de png's de la lista ALFAVIDEOBANK

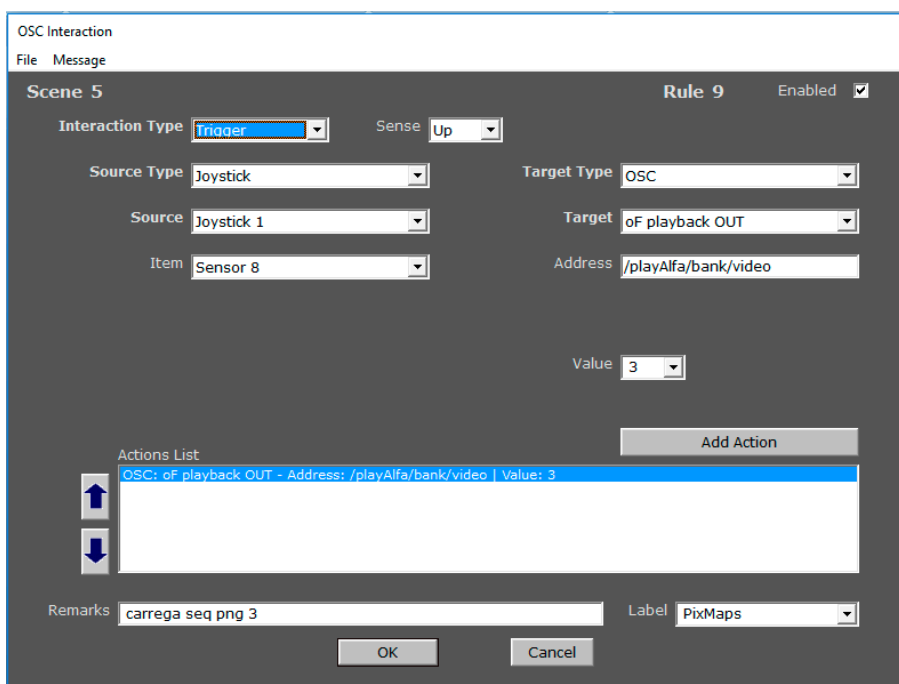
La source es Joystick | Joystick 1 | sensor 6 | que activa OSC | oF playback OUT | con la address: **playAlfa/bank/video /** El value es 1, es decir la segunda carpeta de la lista de ALFAVIDEOBANS. Esta versión de PIXMAPS cuenta desde 0.



09 rule 005 SCENE.

Con el sensor 8 del Joystick 1 carga una secuencia de png's de la lista de ALFAVIDEOBANK

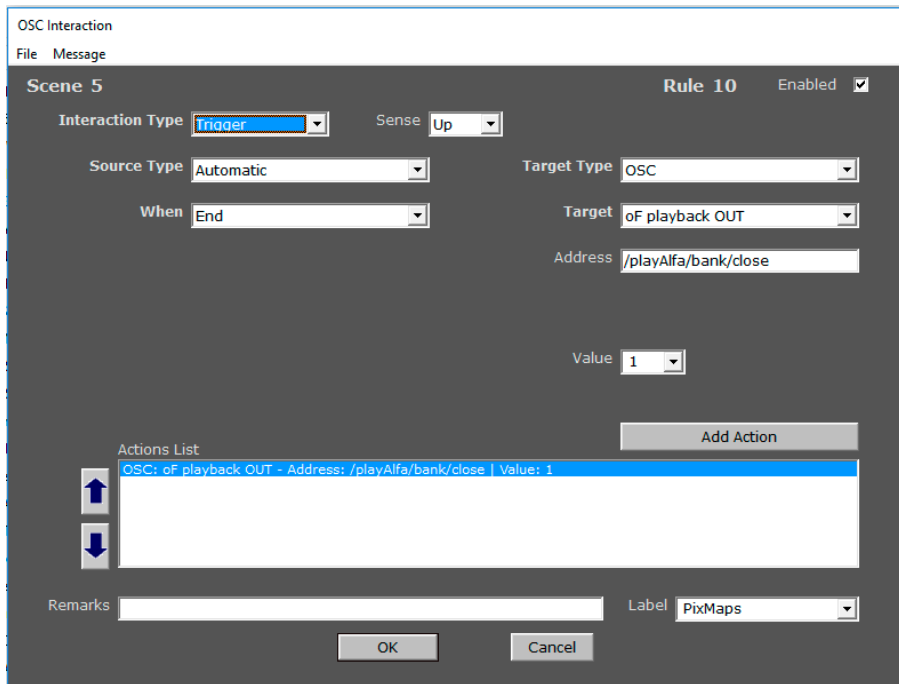
La source es Joystick | Joystick 1 | sensor 6 | que activa OSC | oF playback OUT | con la address: **playAlfa/bank/video** / El value es 3, es decir la cuarta carpeta de la lista de ALFAVIDEOBANS. Esta versión de PIXMAPS cuenta desde 0.



10 rule 005 SCENE.

Cierra Alfavideobank.

La source es Automatic | End | que activa OSC | con la address: **playAlfa/bank/close** descarga el group de las templates, value 1.



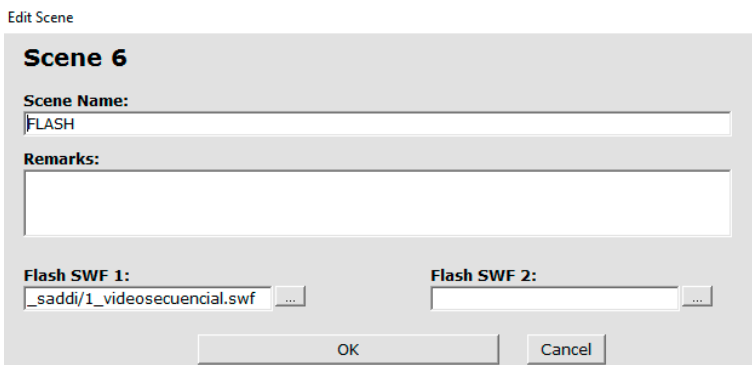
006 FLASH SCENE

Esta escena contiene las reglas para el player de una archivo de video en flv, es decir un vídeo en formato flash.

A diferencia de las reglas anteriores que utilizan a PIXMAPS como target aquí utilizamos otra aplicación gráfica. El hecho de poder poner marcas en las películas nos permite desplazarnos en la línea del tiempo de una marca a otra señalando aquellos puntos de interés que tiene el vídeo. Lo utilizamos mucho para las presentaciones y conferencias. Esta aplicación solo admite el formato de video flv.

GLOBAL RULES							Add rule	
Scenes	Sources	#	Source	Target	Type	Label	Remarks	Description
000 CAMERA	<input type="checkbox"/>	1	AUTO	OSC	TRIG	PART	auto part ->...	Automatic -> OSC Begin
001 VIDEO_PIXMAPS	<input checked="" type="checkbox"/>	2	AUTO	FLA	TRIG		play when st...	Automatic -> Flash Player
002 BANK_VIDEO+PIXMAPS	<input checked="" type="checkbox"/>	3	AUTO	POL	TRIG		switch windo...	Automatic -> POL Trigger
003 DIR+TEMPLATE+GROUPS	<input type="checkbox"/>	4	FLA	FLA	GATE		ANDROID to...	Flash Player -> Flash Playe
004 CAMERA_ALFAVIDEOBANK	<input checked="" type="checkbox"/>	5	FLA	FLA	GATE		ANDROID st...	Flash Player -> Flash Playe
005 BANK+TEMPLATES+SEQ_PNG	<input checked="" type="checkbox"/>	6	FLA	FLA	GATE		ANDROID st...	Flash Player -> Flash Playe
006 FLASH	<input checked="" type="checkbox"/>	7	FLA	FLA	GATE		ANDROID st...	Flash Player -> Flash Playe
007 PART CHANGING	<input checked="" type="checkbox"/>	8	FLA	FLA	GATE		ANDROID st...	Flash Player -> Flash Playe
008 POL_COUNTER_VARIABLES	<input checked="" type="checkbox"/>	9	TOT	FLA	GATE	flash	stream_see...	Joystick -> Flash Player G
009 LIGHTS CONTROL (D-LIGHT)	<input checked="" type="checkbox"/>	10	TOT	FLA	GATE	flash	stream_see...	Joystick -> Flash Player G
010 END	<input checked="" type="checkbox"/>	11	TOT	FLA	GATE	flash	stream_pre...	Joystick -> Flash Player G
	<input checked="" type="checkbox"/>	12	TOT	FLA	GATE	flash	stream_next...	Joystick -> Flash Player G
	<input checked="" type="checkbox"/>	13	TOT	FLA	GATE	flash	toggleplayp...	Joystick -> Flash Player G

A diferencia de la plataforma de gráficos PIXMAPS, en flash llamamos al archivo que vamos a ejecutar a la entrada de la scene desde el submenú Scene. Para abrirlo doble clic sobre la 0006 FLASH de la lista de Scenes. Observa que hay dos casillas en la parte inferior con el nombre de Flash SWF 1 y Flash SWF 2, en nuestra performance la casilla que llama a los archivos flash es la Flash SWF 1, recuerda que la segunda es para ANDROID.

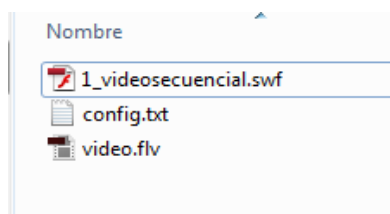


En esta casilla escribiremos el nombre de la carpeta **_saddi/** y a continuación **1_videosecuencial.swf**.

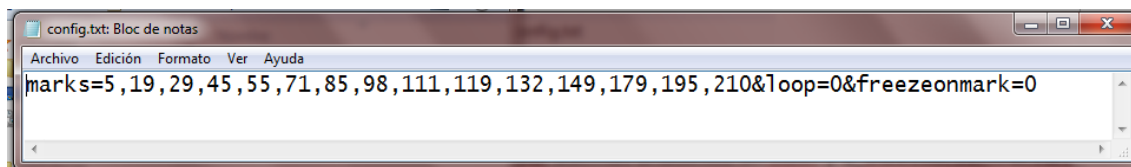
Verás que en el interior de la carpeta donde tengas los gráficos que se encuentra en, en nuestro ejemplo:

D:\graphics_performances\systorg\flash

esta la carpeta **_saddi** dentro de ella hay tres archivos **1_videosecuencial.swf**, **config.txt** y **video.flv**. Esta es la imagen:



Observa que al llamar a la carpeta **_saddi** además añadimos un slash y **1_videosecuencial.swf**. Este swf es el que reproduce el **video.flv**. En realidad este vídeo es el de **_saddi.flv** pero al copiarlo aquí le hemos puesto un nombre genérico para que el swf pueda reproducirlo. De este modo no tendremos que publicar el swf cada vez con un nombre distinto. El otro archivo es **config.txt** nos permite determinar las marcas de los vídeos y si este se reproduce en loop o no, veremos en el párrafo siguiente como funciona. Estamos hablando de que el interior de esta carpeta y todas las que queramos añadir para reproducir vídeos en formato flv en POL, son genéricas y tienen siempre los mismos nombres, aunque estos nombres sean contenidos diferentes. Lo que define el contenido es el nombre de la carpeta, en este caso **_saddi**. Si abrimos el archivo de texto **config.txt** veremos esto:



En el tenemos **marks=** que nos permite avanzar el vídeo según las marcas que pongamos. Las marcas están en segundos y van siempre con una coma a excepción de la última que lleva este signo **&** y sirve para cerrar las marcas.

Tenemos **loop= 0** que reproduce el vídeo sin loop, y con **loop= 1** que lo reproduce con loop, cerramos con **&**.

freezeonmark=0 así va de una marca a otra y no se para con **freezeonmark=1** va de una marca a otra parándose. Como veremos hay otras formas de hacer pausa. Te aconsejamos dejar el **freezeonmark=0** y utilizar la pausa que veremos en una próxima rule.

TABLA VARIABLES para CONTROL del REPRODUCTOR de FLV en FLASH

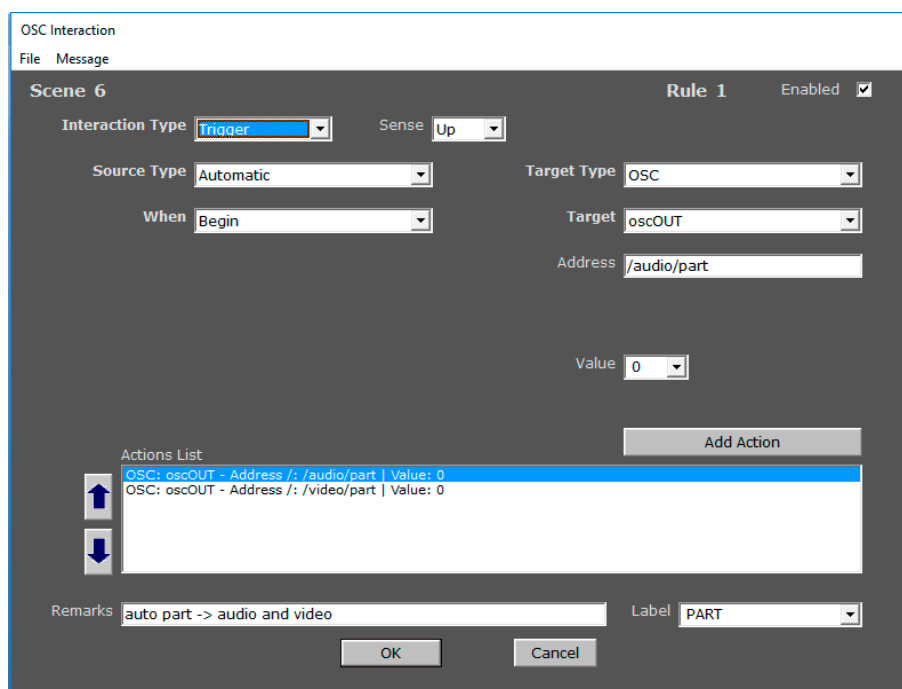
Nombre variable	Para que sirve
stream_play	Activar el vídeo
steam_toggleplaypause	Pausa
steam_seekforward	Avance rápido
steam_seekbackward	Retroceso rápido
steam_prevmark	Ir a Marca Previa
steam_nextmark	Ir a próxima Marca.

1 rule 006 SCENE.

Como cargar una audio part en el cambio de parte.

En el sistema descrito en los manuales de POL Setup y POL Control hemos definido un laptop dedicado al audio. No tratamos aquí este aspecto pero esta rule nos habla de como lanzar un tema en la aplicación de audio PD que solemos utilizar. Los mensajes de OSC se hicieron para esta aplicación. Es muy probable que si utilizas otra aplicación de audio tengas que utilizar otros mensajes OSC.

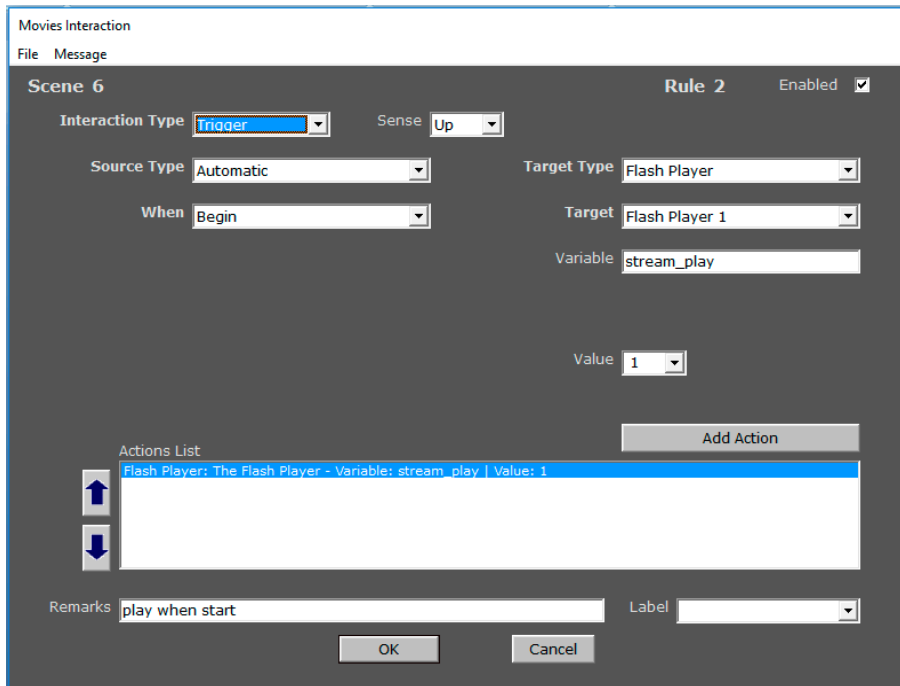
La source es Trigger | Up | Automatic | Begin | que activa OSC | oscOUT | con la address: **/audio/part/** El value es 0, es decir no cargará ningún tema en PD audio. Si pusiéramos un número nos cargaría la parte que correspondiera a ese número.



2 rule 006 SCENE.

Como poner en play el video flv que hemos elegido en el submenú Scene en el cambio de parte.

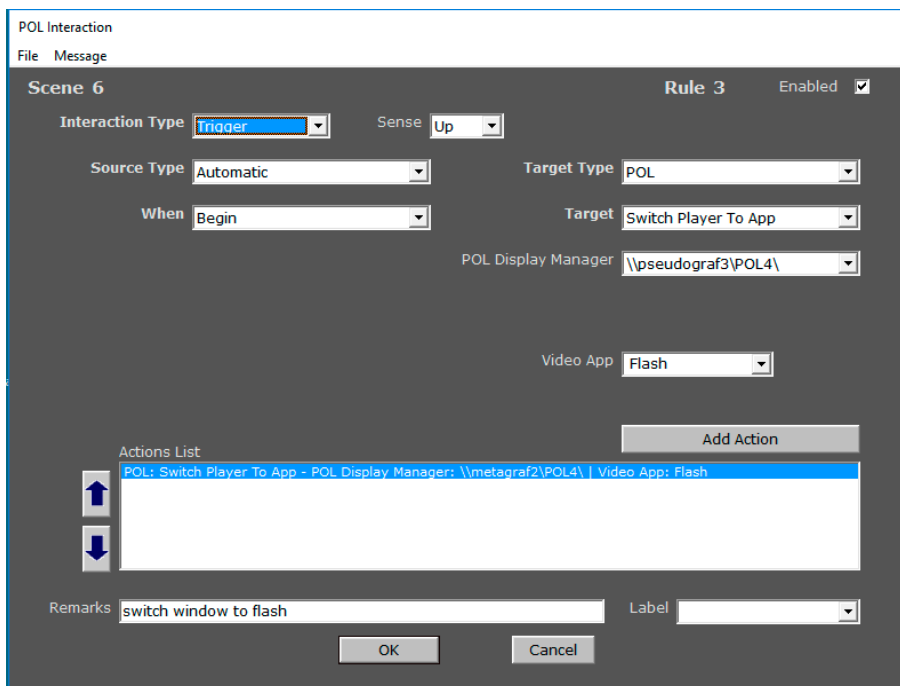
La source es Trigger | Up | Automatic | Begin | que activa Flash Player | Flash Player 1 | con la variable: **stream_play/** El value es 1. Así el vídeo arranca al cambiar de parte.



3 rule 006 SCENE.

Como poner la aplicación Flash en primer termino del laptop de gráficos en el cambio de parte con POL Display Manager. Esta rule es imprescindible cuando utilices Flash y PIXMAPS, o cualquier otra aplicación de graficos.

La source es Trigger | Up | Automatic | Begin | que activa POL | witch Player To App|con Pol Display Manager: \\pseudograf3\POL4\ Observa que aquí va la ruta donde esta el programa de POL Display Manager y que tendrás que adaptarla a tu propia ruta.

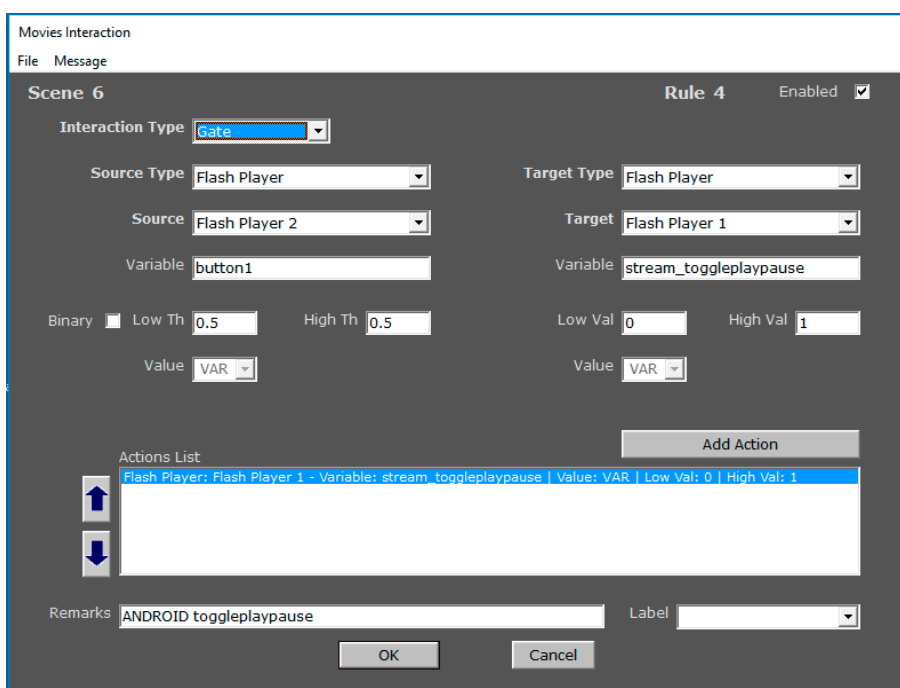


4 rule 006 SCENE.

Definimos el boton de ANDROID con el cual vamos hacer pause del vídeo que estamos reproduciendo.

Aquí hemos mantenido ANDORID como interfaz de entrada. Recuerda que puedes configurar un dispositivo Android, tableta o teléfono, y utilizarlos como interfaz. Las próximas 4 rules explican como utilizar ANDROID. Hay un manual de como configurar la aplicación Movieplayer AIR, la responsable de ANDORID, en: systorg.hangar.org

La source es Gate |Flash Player | Flash Player 2 (El 2 es el Flash Player que hemos reservado para ANDORID) | button1 | con los valores Low Th 0.5 | Hight Th 0.5|con el target Flash Player| Flash Player 1 es decir el flash que esta reproduciendo la movie flv |con la variable **steam_toggleplaypause** | con el Low Val 0| High Val 1. Ni en una columna ni en la otra se utiliza la value. La variable **steam_toggleplaypause** para la pausa del vídeo funciona como toggle, es decir cuando pulsas por primera vez se para y cuando pulsas de nuevo se activa.

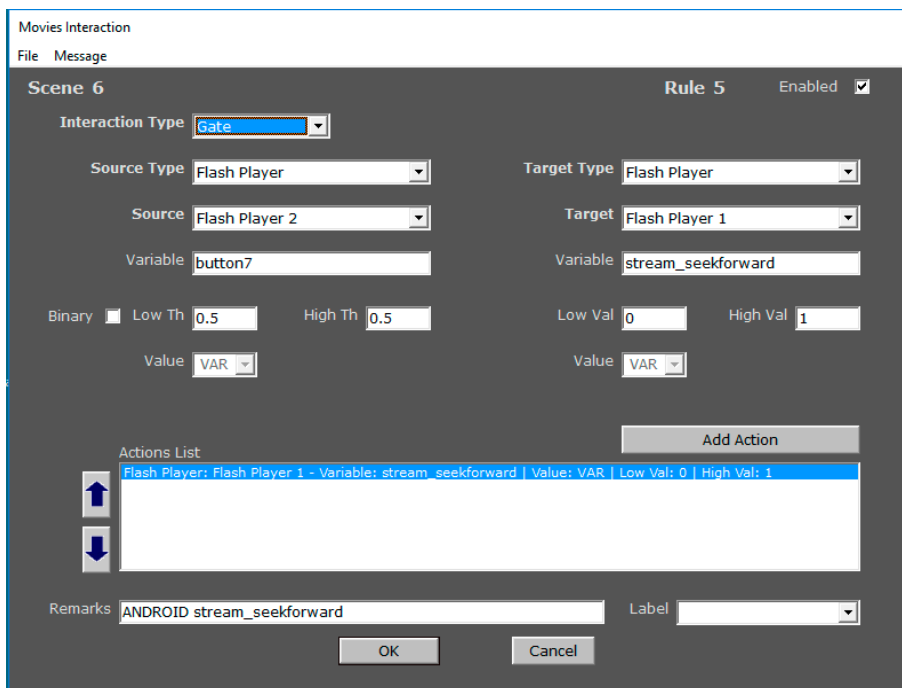


5 rule 006 SCENE.

Definimos el boton de ANDROID con el cual vamos hacer forward, avance rápido, del vídeo que estamos reproduciendo.

Aquí utilizamos ANDROID como interfaz de entrada. Hay un manual de como configurar la aplicación Movieplayer AIR, la responsable de ANDORID, en: systorg.hangar.org

La source es Gate |Flash Player | Flash Player 2 (El 2 es el Flash Player que hemos reservado para ANDORID) | button7 | con los valores Low Th 0.5 | Hight Th 0.5|con el target Flash Player| Flash Player 1 es decir el flash que esta reproduciendo la movie flv |con la variable **steam_seekforward** | con el Low Val 0| High Val 1. Ni en una columna ni en la otra se utiliza la value. La variable **steam_seekforward** acelera el vídeo mientras mantienes pulsado el botón.

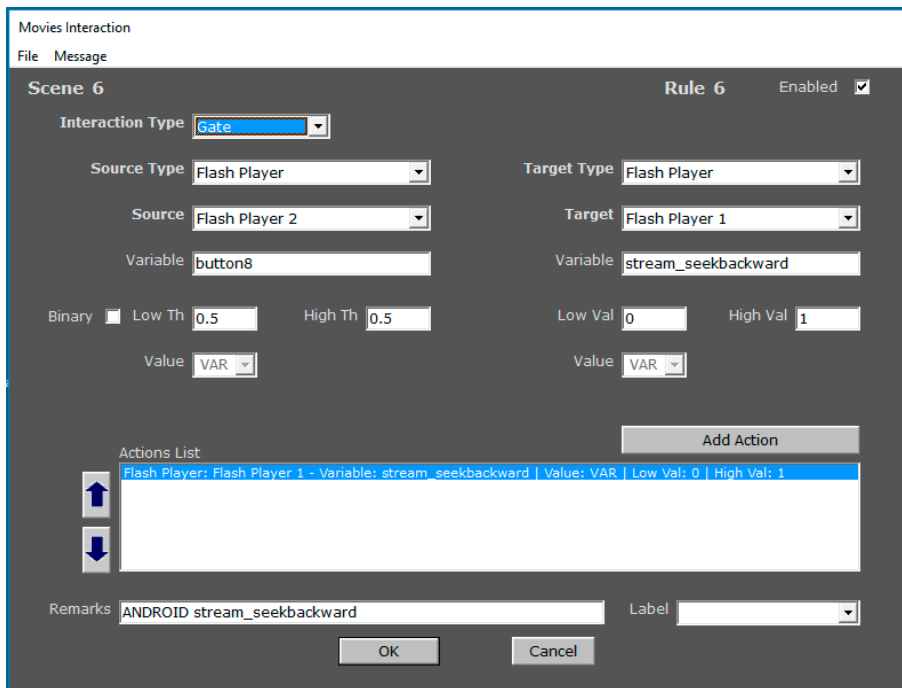


6 rule 006 SCENE.

Definimos el boton de ANDROID con el cual vamos hacer backward, retroceso rápido, del vídeo que estamos reproduciendo.

Aquí utilizamos ANDROID como interfaz de entrada. Hay un manual de como configurar la aplicación Movieplayer AIR, la responsable de ANDORID, en: systorg.hangar.org

La source es Gate | Flash Player | Flash Player 2 (El 2 es el Flash Player que hemos reservado para ANDORID) | button8 | con los valores Low Th 0.5 | Hight Th 0.5|con el target Flash Player| Flash Player 1 es decir el flash que esta reproduciendo la movie flv |con la variable **steam_seekbackward** | con el Low Val 0| High Val 1. Ni en una columna ni en la otra se utiliza la value. La variable **steam_seekbackward** retrocede el vídeo mientras mantienes pulsado el botón.

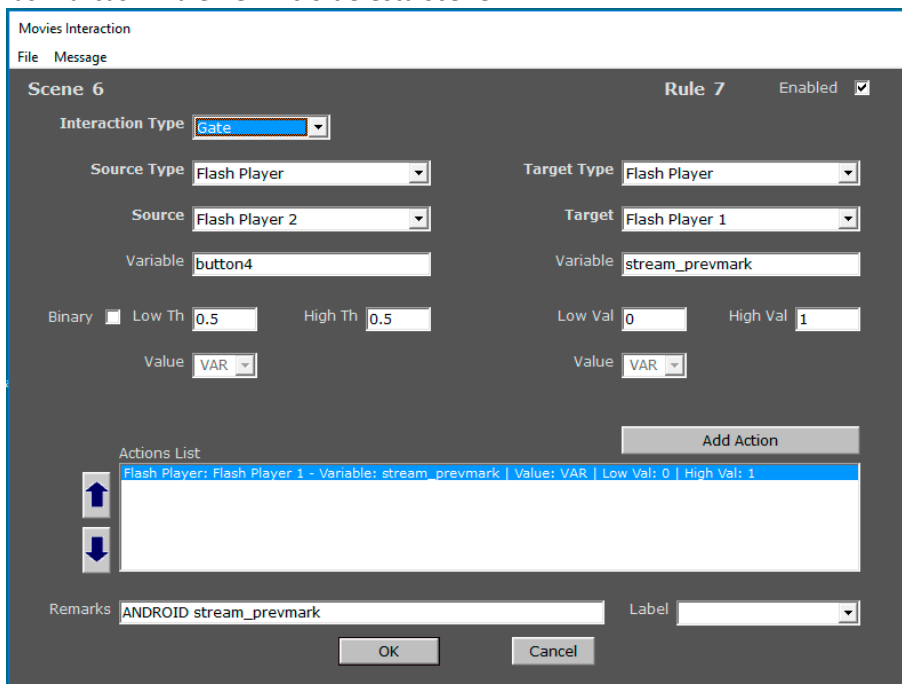


7 rule 006 SCENE.

Definimos el boton de ANDROID con el cual vamos ha ir la marca previa del vídeo que estamos reproduciendo.

Aquí utilizamos ANDROID como interfaz de entrada. Hay un manual de como configurar la aplicación Movieplayer AIR, la responsable de ANDORID, en: systorg.hangar.org

La source es Gate | Flash Player | Flash Player 2 (El 2 es el Flash Player que hemos reservado para ANDORID) | button4 | con los valores Low Th 0.5 | Hight Th 0.5|con el target Flash Player| Flash Player 1 es decir el flash que esta reproduciendo la movie flv |con la variable **steam_prevmark** | con el Low Val 0| High Val 1. Ni en una columna ni en la otra se utiliza la value. La variable **steam_prevmark** nos lleva a la marca anterior del vídeo. Como configurar las marcas mira en el inicio de esta scene.

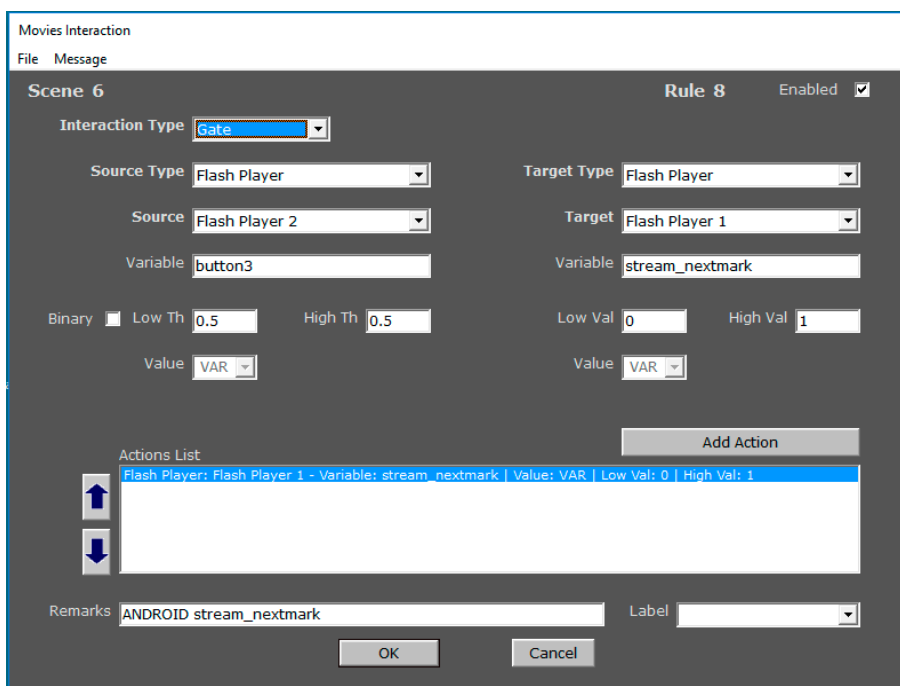


8 rule 006 SCENE.

Definimos el boton de ANDROID con el cual vamos ha ir a la marca próxima del vídeo que estamos reproduciendo.

Aquí utilizamos ANDROID como interfaz de entrada. Hay un manual de como configurar la aplicación Movieplayer AIR, la responsable de ANDORID, en: systorgy.hangar.org

La source es Gate | Flash Player | Flash Player 2 (El 2 es el Flash Player que hemos reservado para ANDORID) | button3 | con los valores Low Th 0.5 | Hight Th 0.5|con el target Flash Player| Flash Player 1 es decir el flash que esta reproduciendo la movie flv |con la variable **steam_nextmark** | con el Low Val 0| High Val 1. Ni en una columna ni en la otra se utiliza la value. La variable **steam_nextmark** nos lleva a la próxima marca del vídeo. Como configurar las marcas mira en el inicio de esta scene.

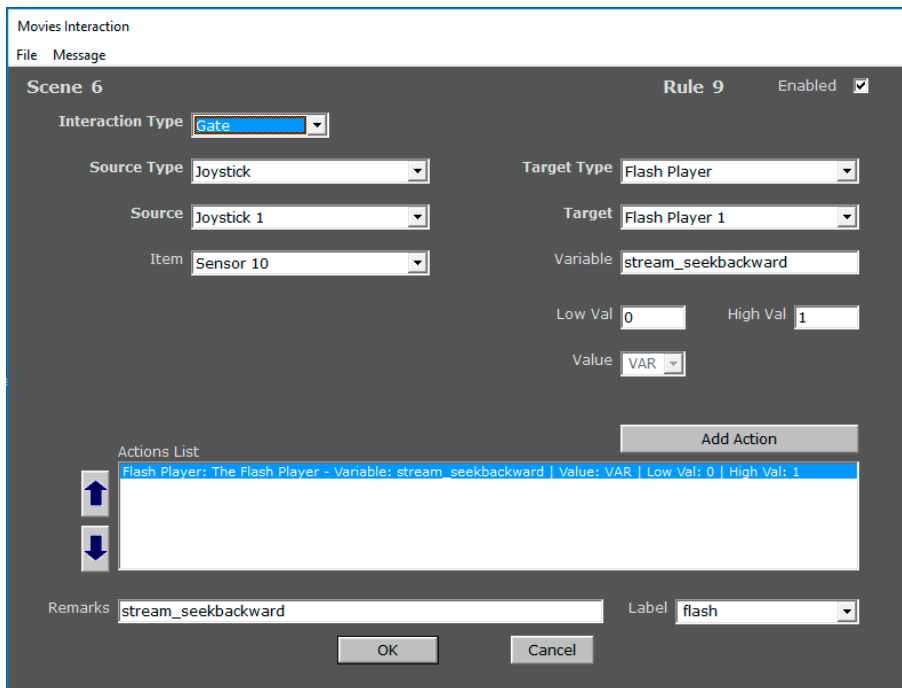


9 rule 006 SCENE.

Definimos con el sensor de la bahía del Joystick 1 como vamos hacer un backward, retroceso rápido, del vídeo que estamos reproduciendo.

Repetiremos las ordenes anteriores de pausa, avance rápido, retroceso rápido, próxima marca y marca anterior con una nueva interfaz, la que hayamos configurado con la bahía del joystick 1. Recuerda que puede ser un simplemente un joystick.

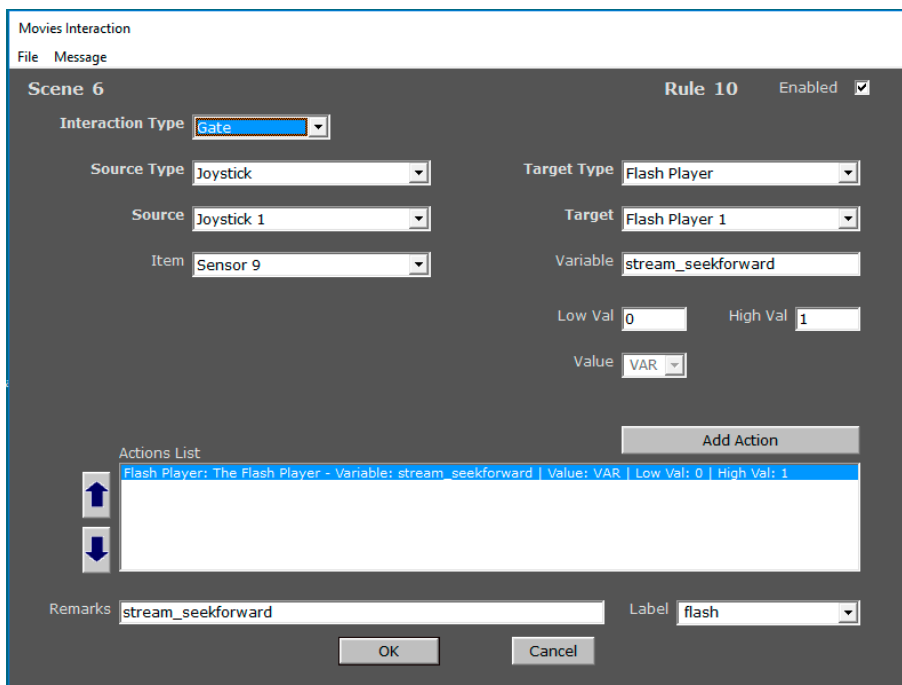
La source es Gate | Joystick | Joystick 1 | Sensor 10 | con los valores Low Th 0.5 | Hight Th 0.5|con el target Flash Player| Flash Player 1 |con la variable **steam_seekbackward** | con el Low Val 0| High Val 1. Ni en una columna ni en la otra se utiliza la value. La variable **steam_seekbackward** Hace el retroceso rápido mientras mantienes pulsado el sensor.



10 rule 006 SCENE.

Definimos con el sensor de la bahía del Joystick 1 como vamos hacer un forward, avance rápido, del vídeo que estamos reproduciendo.

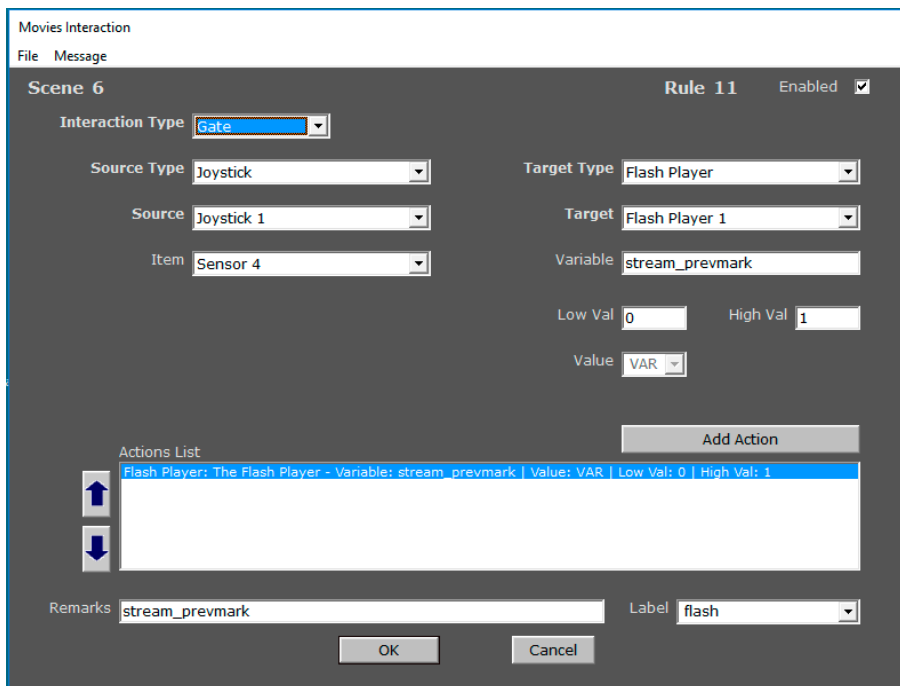
La source es Gate | Joystick | Joystick 1 | Sensor 9 | Flash Player 1 | con la variable **steam_seekforward** | con el Low Val 0 | High Val 1. No se utiliza la value. La variable **steam_seekforward** hace el avance rápido mientras mantienes pulsado el sensor.



10 rule 006 SCENE.

Definimos con el sensor de la bahía del Joystick 1 como vamos como ir al marca previa del vídeo que estamos reproduciendo.

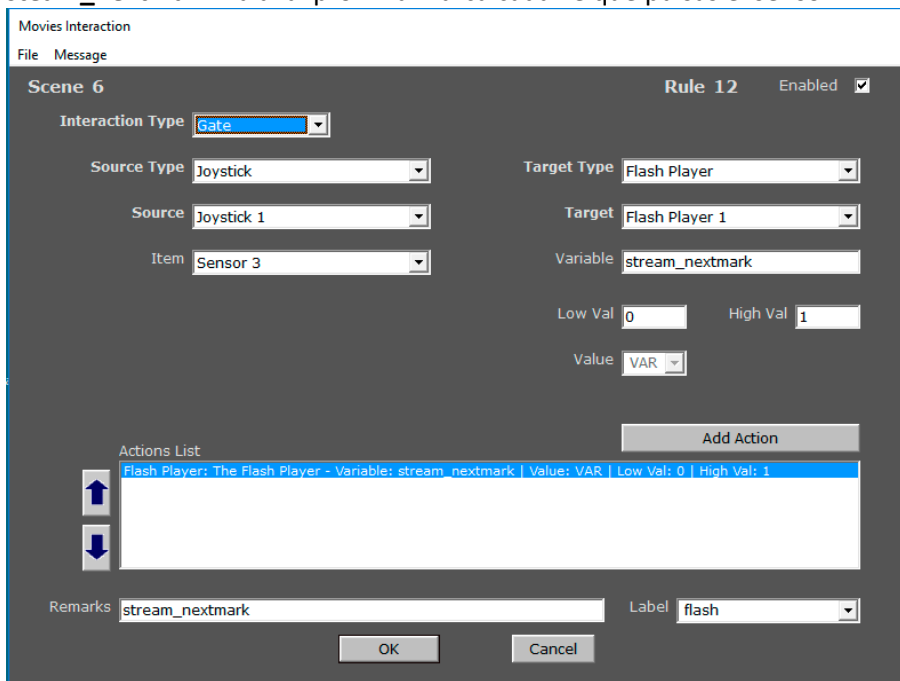
La source es Gate | Joystick | Joystick 1 | Sensor 4 | con el target Flash Player| Flash Player 1 | con la variable **steam_prevmark** | con el Low Val 0 | High Val 1. No se utiliza la value. Variable **steam_prevmark** va a la marca anterior cada ve que pulsas el sensor.



12 rule 006 SCENE.

Definimos con el sensor de la bahía del Joystick 1 como vamos como ir a la próxima marca del vídeo que estamos reproduciendo.

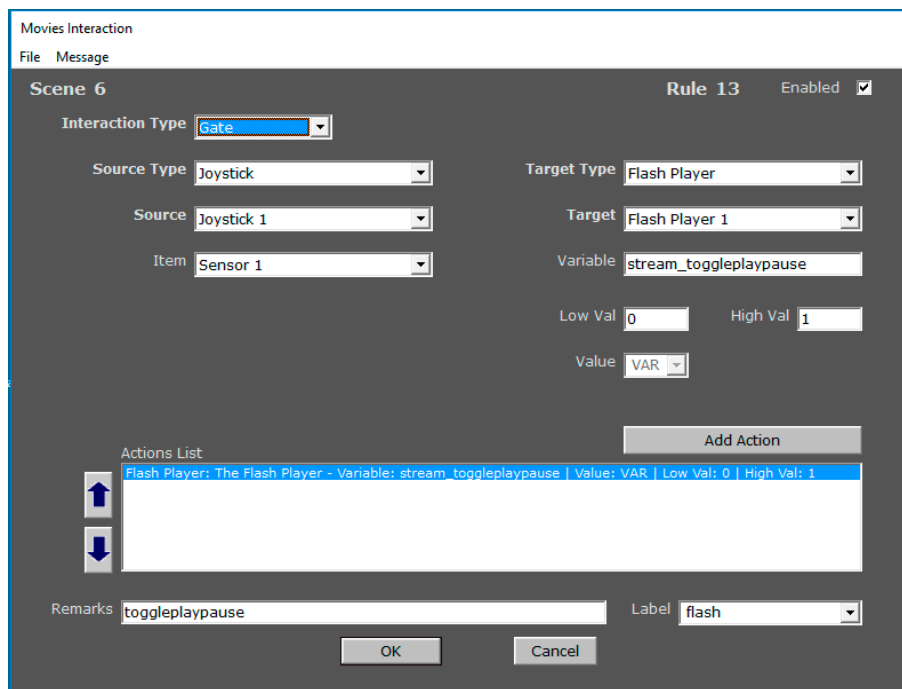
La source es Gate | Joystick | Joystick 1 | Sensor 3 | con el target Flash Player| Flash Player 1 | con la variable **steam_nextmark** | con el Low Val 0 | High Val 1. No se utiliza la value. Variable **steam_nextmark** va a la próxima marca cada ve que pulsas el sensor .



13 rule 006 SCENE.

Definimos con el sensor de la bahía del Joystick 1 como vamos pausa del vídeo que estamos reproduciendo.

La source es Gate | Joystick | Joystick 1 | Sensor 1 | con el target Flash Player| Flash Player 1 | con la variable **steam_toggleplaypause** | con el Low Val 0 | High Val 1. No se utiliza la value. Variable **steam_toggleplaypause** para la pausa del vídeo, funciona como toggle, es decir cuando pulsas por primera vez se para y cuando pulsas de nuevo se activa.



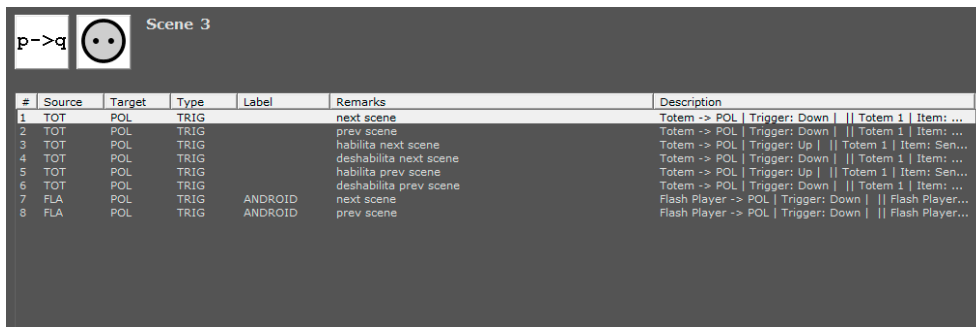
007 PART CHANGING.

Se muestra como realizar una acción, en este caso cambiar de scene, con dos sensores como cuando se utiliza un embrague de automóvil para cambiar una marcha.

Hemos escrito esta scene para acotar el procedimiento de cambiar de parte y sus rules, y con ello facilitar su comprensión. En realidad estas rules deberían formar parte de las Global Rules ya que por su condición jerárquica es desde ahí desde donde se controla el avance y retroceso de las scenes.

En esta scene enseñamos como cambiar de scene y a hacerlo en hacia adelante o hacia atrás. Como ya vimos en la tabla de los target de POL existen dos variables **Next Part** y **Previous Part** que nos permiten respectivamente avanzar o retroceder en el menú de scenes.

Utilizamos una forma compleja de hacerlo que heredamos de el dreskeleton Joysdreske (ver glosario para estos términos) y que consiste en utilizar dos sensores para realizar una sola acción.



#	Source	Target	Type	Label	Remarks	Description
1	TOT	POL	TRIG		next scene	Totem -> POL Trigger: Down Totem 1 Item: ...
2	TOT	POL	TRIG		prev scene	Totem -> POL Trigger: Down Totem 1 Item: ...
3	TOT	POL	TRIG		habilita next scene	Totem -> POL Trigger: Up Totem 1 Item: Sen...
4	TOT	POL	TRIG		deshabilita next scene	Totem -> POL Trigger: Down Totem 1 Item: ...
5	TOT	POL	TRIG		habilita prev scene	Totem -> POL Trigger: Up Totem 1 Item: Sen...
6	TOT	POL	TRIG		deshabilita prev scene	Totem -> POL Trigger: Down Totem 1 Item: ...
7	FLA	POL	TRIG	ANDROID	next scene	Flash Player -> POL Trigger: Down Flash Player...
8	FLA	POL	TRIG	ANDROID	prev scene	Flash Player -> POL Trigger: Down Flash Player...

1ª rule. 007 SCENE. PART CHANGING.

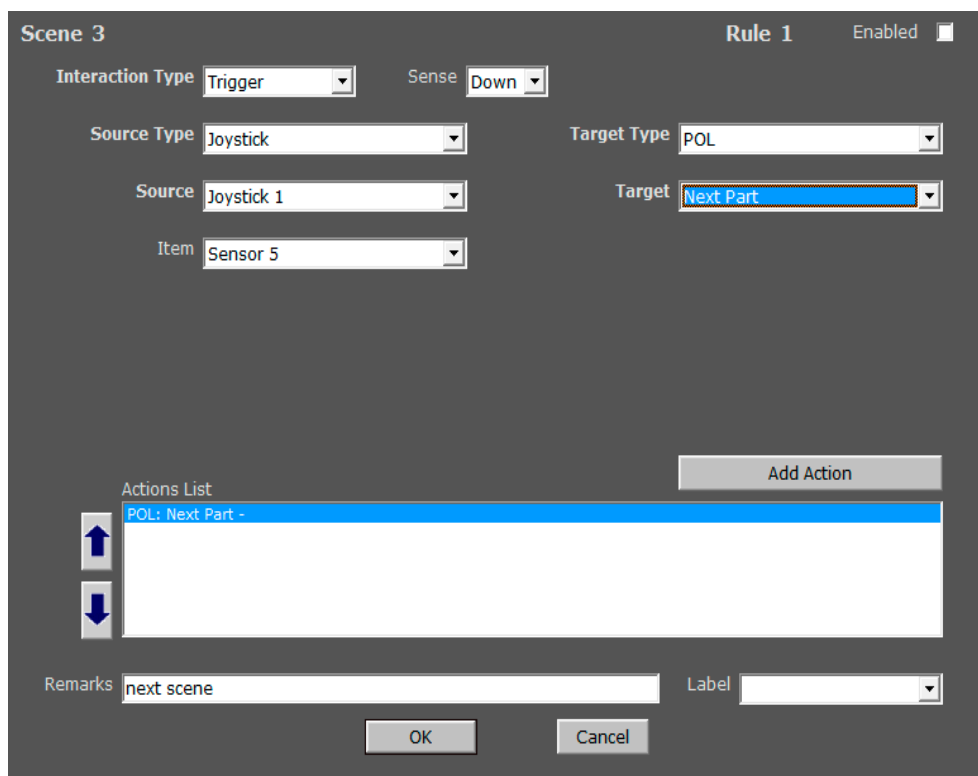
Cambiar el Target Previous Part , scene previa, con dos sensores.

Elegimos el sensor 5 del Joystick1 que en nuestro ejemplo es un interruptor de posición del Joydreske, para ello: Trigger | Down | Joystick | Joystick 1 | Sensor 5, y Target Type POL | Next Part.

Observa que **esta rule no está activa ya que la casilla del Enabled no está señalada**, significa que hasta que no la activemos a través del POL esta rule no funcionará.

Observa también que hemos señalado el Trigger en Down, se trata de leer en el momento en el que el pulsador deja de estar pulsado, como verás para completar esta acción utilizamos el sensor 7 del joystick1, este es un interruptor.

Decidimos hacer el cambio de parte con dos sensores para evitar que pulsando un solo sensor por error cambiase de parte. En este caso esta orden tienen carácter ergonómico de modo que nuevo el interruptor 5 en ON y aprieto el pulsador 7 y después vuelvo el interruptor 5 al inicio (Down) y la orden se ejecuta.



Scene 3 Rule 1 Enabled

Interaction Type: Trigger Sense: Down

Source Type: Joystick Target Type: POL

Source: Joystick 1 Target: Next Part

Item: Sensor 5

Actions List

POL: Next Part -

Remarks: next scene Label: []

OK Cancel

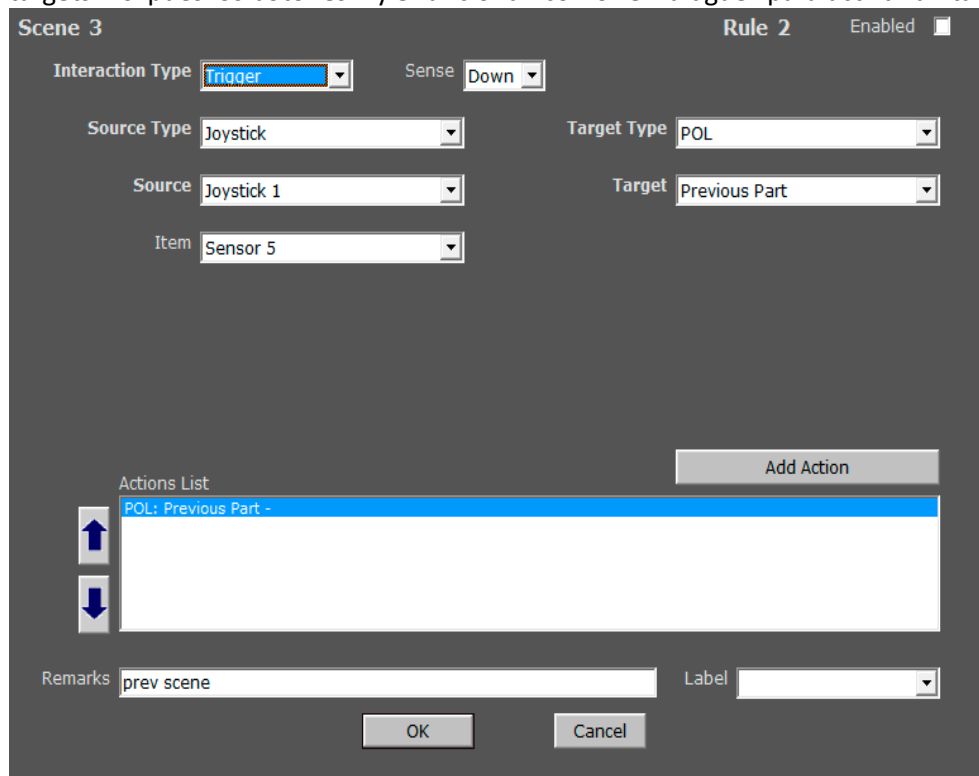
2ª rule. 007 SCENE. PART CHANGING.

Cambiar el Target Previous Part , scene previa, con dos sensores.

Elegimos como en la rule anterior el sensor 5 observa que es igual y solo cambia el target de POL **Previous Part**, es decir scene anterior. Enabled sigue sin estar activo. Como veremos la

orden se ejecuta con el botón 8 del Joydreske.

Esto significa que el pulsador 5 se utiliza en las dos rules, y en consecuencia para los dos targets. Así pues los botones 7 y 8 funcionan como “embrague” para activar un target u otro.



3ª rule. 007 SCENE. PART CHANGING.

Habilita la 1ª rule y hace efectivo el target Next Part.

Tenemos Trigger : Up | Josystick | Joystick 1 | Sensor 7 ahora la Enablet esta marcada y por tanto activa. El Target Type POL | Enable Event. El Event # es 1, esto significa que con el botón 7 del joydreske, siempre siguiendo nuestro ejemplo, activamos la rule 1. Observa que bajo la casilla event un resumen de la rule 1:

POL | Trigger:Down || Totem 1 | Item sensor 5* || POL: Next part –

Recuerda que Totem es otra acepción de Joystick en POL.

Verás que la casilla de Actions List contiene una segunda rule

POL: Disable Event – Event #: 2

Esta rule sirve para que con la misma acción desactivemos el target de Previus Part. Esta rule es solo como prevención.

Recuerda que es importante el orden en que colocamos los events en la Actions list y que el primero, en este caso:

POL | Trigger:Down || Totem 1 | Item sensor 5* || POL: Next part, será el primero de la lista.

4ª rule. 007 SCENE. PART CHANGING.

Deshabilita la 1ª rule y hace elimina el target Next Part

El hecho de utilizar 2 sensores nos obliga a ser cuidadosos con habilitar y deshabilitar las rules. Esta Rule deshabilita el target Next Part. Según la ergonomía del Joydreske, el botón 7 es un pulsador con muelle que cuando dejas de pulsarlo vuelve a su posición inicial, en POL Trigger: Down, por lo que ejecuta la rule que a continuación describimos.

Tenemos Trigger: Down | Joystick | Joystick 1 | Sensor 7 ahora la Enablet esta marcada y por tanto activa. El Target Type POL | Diable Event. El Event # es 1, esto significa que con el botón 7 del joydreske de vuelta a su estado de reposo, desactivamos la rule 1.

5ª rule. 007 SCENE. PART CHANGING.

Habilita la 2ª rule y en la que el target Previous Part se activa.

Tenemos Trigger: Up | Joystick | Joystick 1 | Sensor 8 ahora con el Enablet activo. El Target Type POL | Enable Event. El Event # es 2, esto significa que pondrá, junto al sensor 5, en marcha la rule 2, que vemos escrita bajo la casilla Event #

Totem-> POL | Trigger: Down | || Totem 1 | Item: sensor 5* || POL: Previous Part –

Como en la 3ª rule la casilla de Actions List contiene una segunda rule

POL: Disable Event – Event #: 1

Esta rule sirve para que con la misma acción desactivemos el target de Next Part. Esta rule funciona solo como prevención.

Scene 3 Rule 5 Enabled

Interaction Type: Trigger Sense: Up

Source Type: Joystick Target Type: POL

Source: Joystick 1 Target: Enable Event

Item: Sensor 8 Event #: 2

Totem -> POL | Trigger: Down | || Totem 1 |
Item: Sensor 5 * || POL: Previous Part -

Actions List

- POL: Enable Event - Event #: 2
- POL: Disable Event - Event #: 1

Remarks: habilita prev scene Label:

OK Cancel

6ª rule. 007 SCENE. PART CHANGING.

Desabilita la 2ª rule y hace elimina el target Previous Part

Funciona como la 4ª rule. Cambian solo el Item, aquí 8 y el Event # aquí 2.

Scene 3 Rule 6 Enabled

Interaction Type: Trigger Sense: Down

Source Type: Joystick Target Type: POL

Source: Joystick 1 Target: Disable Event

Item: Sensor 8 Event #: 2

Totem -> POL | Trigger: Down | || Totem 1 |
Item: Sensor 5 * || POL: Previous Part -

Actions List

- POL: Disable Event - Event #: 2

Remarks: deshabilita prev scene Label:

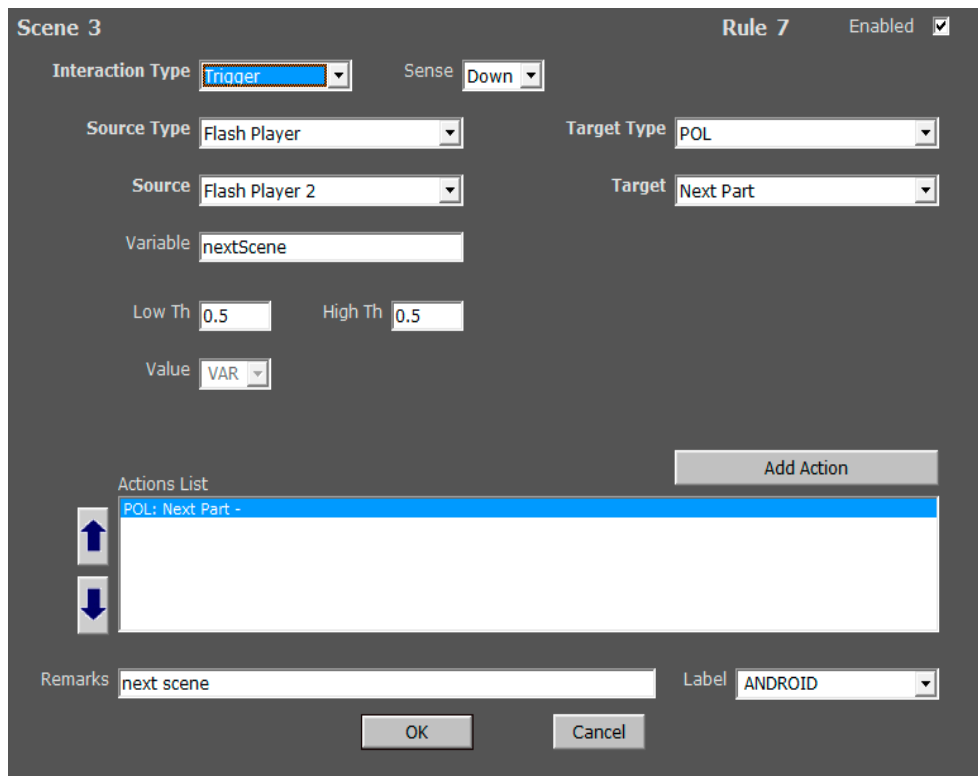
OK Cancel

7ª rule. 007 SCENE. PART CHANGING.

Cambiar el Target Next Part , con la table Android.

Cambiamos con el dispositivo Android aquí como Source type Flash Player 2. Esto se puede ver en detalle en el apartado MovieplayerAIR situado en otro documento. Tanto los botones virtuales como las pantallas se puede configurar en un swf según las necesidades de la performance. Con ello podemos configurar los botones de cambio de parte en una pantalla específica y desplazarnos a ella solo cuando queramos.

Tenemos la Interaction Type Trigger: Down | Flash Player | Flash Player 2 | nextScene | 0.5 / 0.5 , la nextScene es la variable prefigurada ya en MovieplayerAIR el Target Type es POL | Next Part.

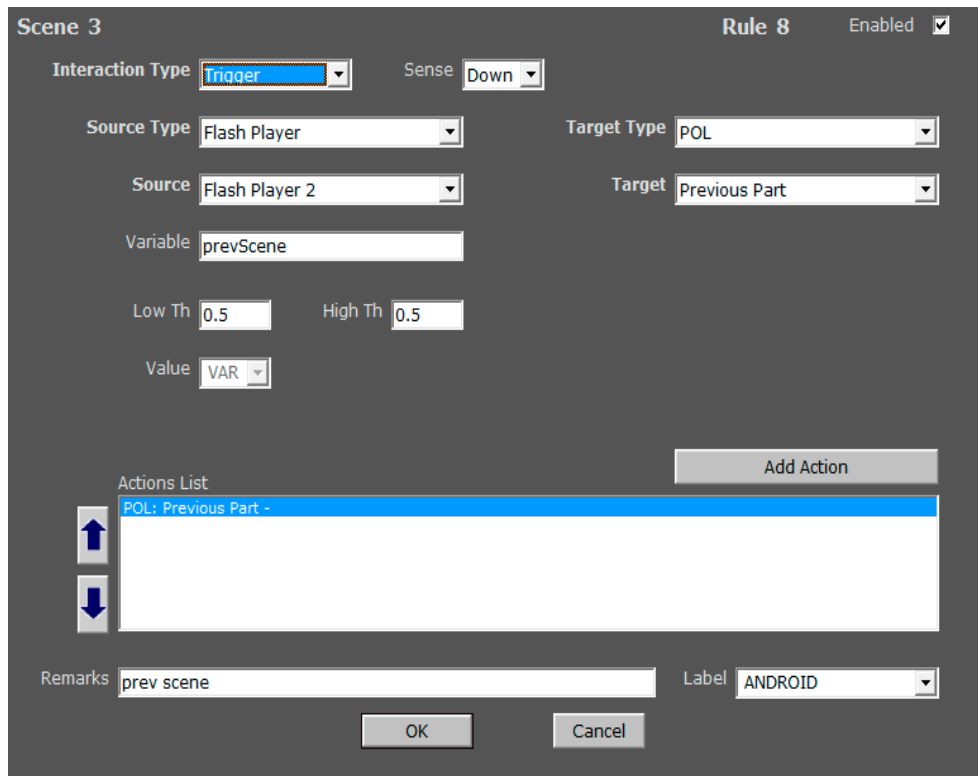


8ª rule. 007 SCENE. PART CHANGING.

Cambiar el Target Next Part , con la tablet Android.

Cambiamos con el dispositivo Android aquí como Source type Flash Player 2. Esto se puede ver en detalle en el apartado MovieplayerAIR situado en otro documento. Tanto los botones virtuales como las pantallas se puede configurar en un swf según las necesidades de la performance. Con ello podemos configurar los botones de cambio de parte en una pantalla específica y desplazarnos a ella solo cuando queramos.

Tenemos la Interaction Type Trigger: Down | Flash Player | Flash Player 2 | nextScene | 0.5 / 0.5 , la nextScene es la variable prefigurada ya en MovieplayerAIR el Target Type es POL | Next Part.



008 SCENE. POL COUNTER VARIABLES

En esta scene vemos algunos ejemplos del COUNTER , contador de POL. Para establecer un contador tenemos que darle un nombre y escribirlo en casilla Var Name, en nuestro ejemplo es: counter_1, pero podríamos escribir otro. Esta es la scene sobre la que aplicaremos los ejemplos de incremento, decremento, reset o ira a.

El contador funciona como un ábaco al que le sumamos o restamos bolas. Utilizaremos esta imagen para explicar el contador. Podemos añadir o sacar bolas asignando esta tarea a sensores. Un sensor será incremental y otro, si así lo deseamos, decremental. Si en un momento tenemos el ábaco en tres bolas y a ese estado de tres le hemos asignado un sonido sonará, si hemos asignado otro sonido en el cuatro al llegar a cuatro sonará... El contador en sí mismo no actúa sobre targets sino que construye el instrumento para poderlo hacer. El sistema de counter introduce una cierta complejidad al sistema pero pueden resultar muy útil para dar versatilidad a la interfaz si disponemos de pocos sensores o queremos utilizar un único sensor para distintos usos.

La Tabla de variables de POL que está en este manual en la sección Menú Rules | 13-Target.

La Scene 001 contiene estas 7 rules que hemos preparado como ejemplo para entender el counter:

#	Source	Target	Type	Label	Remarks	Description
1	AUTO	POL	TRIG	counters	reset counter	Automatic -> POL Trigger: Up Begin POL: Assign Value To Variable - Var Name: counter_1 Value: 0
2	TOT	POL	TRIG	counters	incremental counter - switch 1	Joystick -> POL Trigger: Up Joystick 1 Item: Switch 1 POL: Increment Variable - Var Name: counter_1 Value: 1
3	TOT	POL	TRIG	counters	decremental counter - switch 2	Joystick -> POL Trigger: Up Joystick 1 Item: Sensor 2 POL: Decrement Variable - Var Name: counter_1 Value: 1
4	TOT	POL	TRIG	counters	reset counter - switch 5	Joystick -> POL Trigger: Up Joystick 1 Item: Sensor 5 POL: Assign Value To Variable - Var Name: counter_1 Value: 0
5	AUTO	POL	TRIG	counters	counter when arrives at 10 gets part 1	Automatic -> POL Trigger: Up POL Variable Min: 10 Max: 10 POL: Go To Scene - Scene: 1 CAMERA
6	AUTO	OSC	TRIG	counters	when arrives at 4 send OSC variable	Automatic -> OSC Trigger: Up POL Variable Min: 4 Max: 4 OSC: Lights out - Address: /example Value: 23
7	AUTO	OSC	TRIG	counters	counter when arrives at 3 trigger sampler x	Automatic -> OSC Trigger: Up POL Variable Min: 3 Max: 3 OSC: oscOUT - Address: /audio/sampler1/memory/onoff Value: 1

Observamos que en la columna de las Source cuatro son AUTO, Automáticas, y tres son TOT, es decir joystick Los target cinco son de POL, es decir referidos al propio programa POL y dos a OSC. Y los Type es decir el comportamiento de la interfaz es TRIG, trigger. Veamos las rules una por una:

1 rule. 008 SCENE. POL COUNTER VARIABLES

Como borrar los Counter Variables anteriores en una nueva scene.

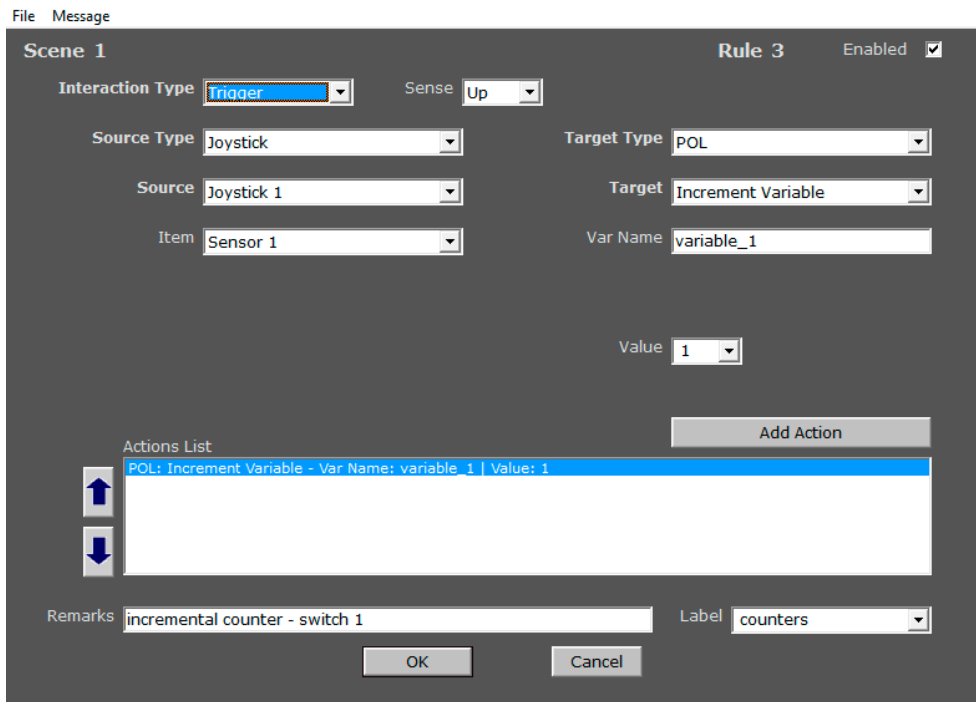
Esta rule nos permite borrar los elementos, bolas virtuales del ábaco, que contenga el counter que hemos designado con la variable counter_1 en el caso que arrastrásemos el mismo counter en scenes anteriores. Se trata de una precaución que en esta performance resulta innecesaria pero que no está de más.

Trigger | Up | Automatic | Begin, es decir la entrada automática a la parte funciona como un sensor trigger que dentro de **POL | assign value To Variable | counter_1 | 0**, que asigna un cero al counter este queda en cero. Como vemos se trata de orden Automatic y se aplicara al entrar a esta escena.

2ª rule. 008 SCENE. POL COUNTER VARIABLES

Como escribir una variable creciente en el Counter de POL.

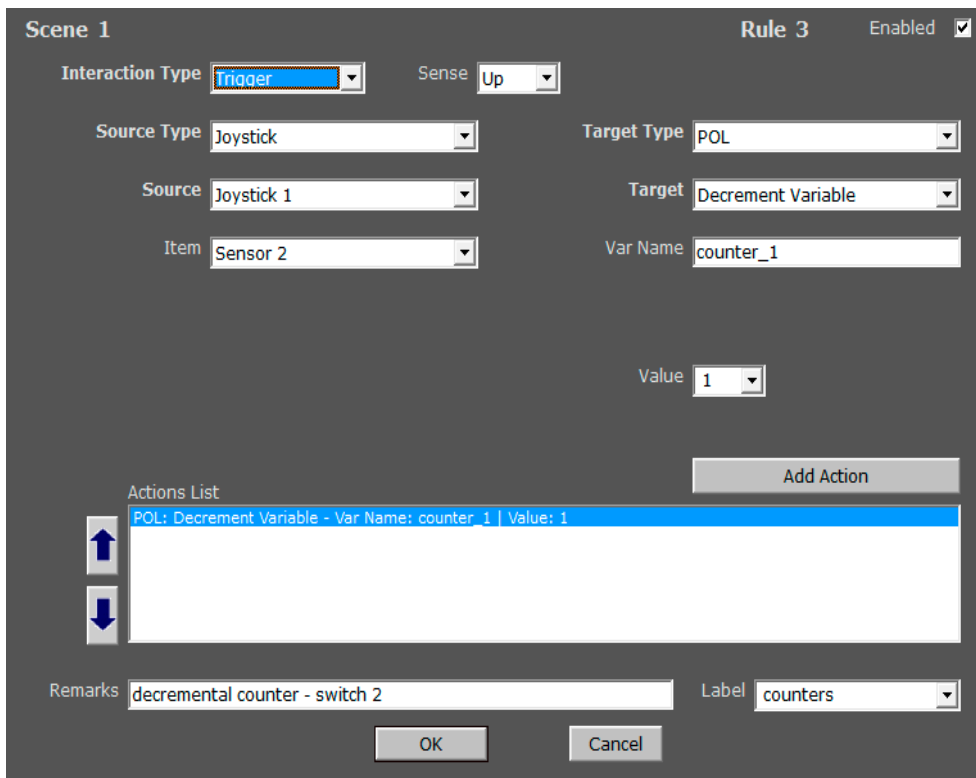
Asignamos una Interaction type, **Trigger | up | joystick 1 | Sensor 1**, recuerda que el sensor uno es on/off. La Target type es POL, **Increment Variable**, Value 1. Esto significa que crece una bola cada vez que activamos el sensor 1 del joystick. Si escribiéramos 2 crecería en múltiplos de dos. Es aconsejable el incremento de uno en uno.



3ª rule. 008 SCENE. POL COUNTER VARIABLES

Como como escribir una variable decreciente.

Asignamos una **Interaction type | Trigger | up | joystick 1 | Sensor 2**, es decir se activa con el sensor dos del Joystick 1. La Target type es **POL : Decrement Variable**, que actúa sobre el **Counter_1**. Según está escrito en la **Value 1**, nuestro ábaco decrece 1 una bola cada vez que activamos el sensor 2 del joystick1. Si la value escribiéramos 2 decrecería múltiplos de 2.



4ª rule. 008 SCENE. POL COUNTER VARIABLES**Como como resetear el counter en un número determinado**

Asignamos una Interaction type, Trigger, up, al joystick 1, Sensor 5. La Target type es POL, **Assign Value To Variable**, siempre en la counter_1. Al escribir en la Value 0 llevamos el contador cero cada vez que activamos el sensor 5 del joystick1 sea cual sea la posición de las bolas del ábaco. Funciona como la 1ª rule de esta scene pero asignada a un sensor.

Scene 1 Rule 4 Enabled

Interaction Type: Trigger Sense: Up

Source Type: Joystick Target Type: POL

Source: Joystick 1 Target: Assign Value To Variable

Item: Sensor 5 Var Name: counter_1

Value: 0

Actions List

POL: Assign Value To Variable - Var Name: counter_1 | Value: 0

Remarks: reset counter - switch 5 Label: counters

OK Cancel

5ª rule. 008 SCENE. POL COUNTER VARIABLES**Como como enviar desde el counter una orden a un target de una de las scenes de esta performance.**

Observad que es esta rule ya partimos de la configuración anterior del counter. Nuestro ábaco añadirá bolas, una a una, cada vez que activemos el sensor 1 del joystick1. Del mismo modo que sacara bolas una a una cada vez que activemos el sensor 2 del joystick1. Tenemos que contar pues las veces que actuamos sobre el sensor para saber en qué número nos encontramos. En el siguiente ejemplo activamos la Interaction Type **Trigger | Automatic | POL Variable | counter_1**, en las casillas que aparecen escribimos dos veces 10, es decir que cuando lleguemos a la bola 10 de nuestro ábaco se activara el target type POL |Go To Scene | 1 CAMERA, es decir ira a la scene uno de esta performance. Observat que esta casilla contiene todas la scenes por lo que podríamos enviarla a cualquiera de ellas.

6ª rule. 008 SCENE. POL COUNTER VARIABLES

Como como enviar desde el counter una orden a un target de una memoria de luz.

En el siguiente ejemplo activamos la Interaction Type **Trigger | Automatic | POL Variable | counter_1**, en cada una de las dos casillas que aparecen escribimos 4, es decir que cuando lleguemos a la bola 4 de nuestro ábaco se activara el target type **OSC | Lights out | example**. Example es el nombre que le hemos dado a la memoria de D lights, este programa es el freeware que utilizamos para el control de luz vía DMX. Observad que en la casilla siguiente escribimos 23 que es el número de esta memoria de luces.

Cuando hayamos activado 4 veces el sensor incremental se disparará la memoria 23 del programa de iluminación.

Scene 1 Rule 6 Enabled

Interaction Type: **trigger** Sense: **Up**

Source Type: **Automatic** Target Type: **OSC**

When: **POL Variable** Target: **Lights out**

Variable: **counter_1** Address: **/example**

Min: **4** Max: **4** Value: **23**

Actions List

- OSC: Lights out - Address: /example | Value: 23

Remarks: **when arrives at 4 send OSC variable** Label: **counters**

OK Cancel

7 rule. 008 SCENE. POL COUNTER VARIABLES

Como como enviar desde el counter una orden a un target sonido.

En el siguiente ejemplo activamos la Interaction Type **Trigger | Automatic | POL Variable | counter_1**, en cada una de las dos casillas que aparecen escribimos 3, es decir que cuando lleguemos a la bola 3 de nuestro ábaco se activara el target type **OSC | oscOUT | /audio/sampler1/memory1/onoff**. En este caso la Address nos muestra la ruta específica del programa PD audio. Aquí pues la value es solo 1 activo, 0 inactivo.

Scene 1 Rule 7 Enabled

Interaction Type: **trigger** Sense: **Up**

Source Type: **Automatic** Target Type: **OSC**

When: **POL Variable** Target: **oscOUT**

Variable: **counter_1** Address: **/audio/sampler1/memory1/onoff**

Min: **3** Max: **3** Value: **1**

Actions List

- OSC: oscOUT - Address: /audio/sampler1/memory1/onoff | Value: 1

Remarks: **counter when arrives at 3 trigger sampler x** Label: **counters**

OK Cancel

009 SCENE. POL COUNTER VARIABLES EXAMPLE VIDEO.

La tercera Scene es un ejemplo práctico para enviar desde el contador varias órdenes a un target, en este caso samplers de audio. La scene se compone de 8 rules 5 de las cuales ya se han descrito en la scene anterior.

#	Source	Target	Type	Label	Remarks	Description
1	AUTO	POL	TRIG	counters	reset counter	Automatic -> POL Trigger: Up Begin POL: As...
2	TOT	POL	TRIG	counters	incremental counter - switch 1	Joystick -> POL Trigger: Up Joystick 1 Item: ...
3	TOT	POL	TRIG	counters	decremental counter - switch 2	Joystick -> POL Trigger: Up Joystick 1 Item: ...
4	AUTO	POL	TRIG	counters	assign var 0 when arrives at 5	Automatic -> POL Trigger: Up POL Variable M...
5	AUTO	OSC	TRIG	counters	counter when arrives at 1 trigger sampler 1	Automatic -> OSC Trigger: Up POL Variable M...
6	AUTO	OSC	TRIG	counters	counter when arrives at 2 trigger sampler 2	Automatic -> OSC Trigger: Up POL Variable M...
7	AUTO	OSC	TRIG	counters	counter when arrives at 3 trigger sampler 3	Automatic -> OSC Trigger: Up POL Variable M...
8	AUTO	OSC	TRIG	counters	counter when arrives at 4 trigger sampler 4	Automatic -> OSC Trigger: Up POL Variable M...

Lo que hemos hecho aquí es copiar las rules de la escena anterior **001 POL COUNTER VARIABLES**, y describir algunos elementos. Haremos una descripción simple de cada rule sin el grafico a excepción de alguna de ellas

- 1ª Rule. Reset counter.** Es igual que la rule 1 de lla anterior Scene. Resetea el contador.
- 2ª Rule. incremental counter - switch 1.** Ídem a la rule 2 scene anterior. Incrementa en valores de 1 el contador.
- 3ª Rule. decremental counter - switch 2.** Ídem a la rule 3 scene anterior. decremanta en valores de 1 el contador.
- 4ª Rule. assign var 0 when arrives at 5.** Ídem a la rule 4 scene anterior. Resetea el contador cuando se activa el sensor 5 del joystick1.
- 5ª Rule y siguientes. counter when arrives at 1 trigger sampler 1.** Ídem a la rule 6 scene anterior. Permite enviar en un número del contador un sampler, de este modo al incrementar el contador que esta en 0 al entrar en la scene en 1 se activará la address: **/audio/sampler1/memory1/onoff** , y con ella el sampler 1, en la siguiente rule la 6ª con el segundo incremento activaremos la address **.../memory2/onoff** y así consecutivamente hasta 4 samplers. Recordad que el 5 incremento nos resetea el counter a 0 y la lista empieza de nuevo, loopeando la serie de 4 samplers cada 5 incrementos.

Scene 2 Rule 5 Enabled

Interaction Type: Trigger Sense: Up

Source Type: Automatic Target Type: OSC

When: POL Variable Target: oscOUT

Variable: variable_1 Address: /audio/sampler1/memory1/onoff

Min: 1 Max: 1 Value: 1

Actions List

- OSC: oscOUT - Address: /audio/sampler1/memory1/onoff | Value: 1

Remarks: counter when arrives at 1 trigger sampler 1 Label: counters

OK Cancel

010 LIGHTS CONTROL (D-LIGHT)

Scene modelo para activar algunas reglas de iluminación activadas a través del Programa D-Light con salida DMX, el protocolo estándar de control de iluminación profesional.

D-Light se puede controlar a través del protocolo OSC. Para ello debes tener instalado el programa en algún ordenador, el programa es multiplataforma y se puede utilizar en el mismo ordenador en que se encuentre Pol Control (si hablamos de una instalación por ejemplo), si trabajamos en una performance, lo más recomendable es utilizar un ordenador distinto para el control de las luces, pues si se desconecta momentáneamente del sistema OSC, puede trabajar en el programa de luces con tranquilidad.

Este es el web site del programa para que podáis descargarlo, en él hay mucha documentación para aprender a utilizarlo.

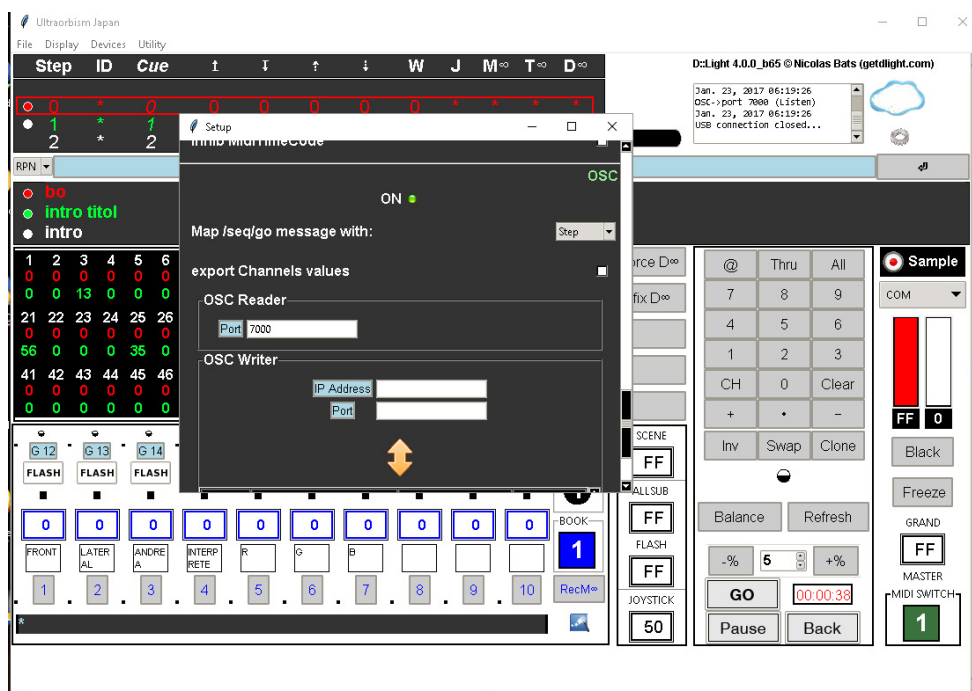
<http://www.getdlight.com/index.php/en/>

En ésta dirección encontrareis información detallada del control OSC del programa, con toda la lista de mensajes de control posibles.

<http://www.getdlight.com/wiki/index.php/OSC>

En éste ejemplo mostramos un ejemplo de control interactivo del programa que os pueden servir para montar una pequeña performance, control de memorias de luces, control de submasters i flashes. No podemos entrar en mostrar como funciona el programa de luces pero si mostramos como interactuar interactivamente con él.

Antes que nada debes activar la lectura de mensajes OSC en DLight, ésta es la pantalla de configuración en el software de luces.



Vamos a configurar algunas reglas para el control del software DLight.

#	Source	Target	Type	Label	Remarks	Description
1	AUTO	OSC	TRIG	lights	lights cue 0	Automatic -> OSC Trigger: Up Begin OSC: Li...
2	TOT	OSC	TRIG	lights	lights cue 1	Joystick -> OSC Trigger: Up Joystick 1 Item: ...
3	TOT	OSC	TRIG	lights	lights cue 2	Joystick -> OSC Trigger: Up Joystick 1 Item: ...
4	TOT	OSC	TRIG	lights	lights cue 3	Joystick -> OSC Trigger: Up Joystick 1 Item: ...
5	TOT	OSC	TRIG	lights	lights cue 4	Joystick -> OSC Trigger: Up Joystick 1 Item: ...
6	TOT	OSC	GATE	lights	switch 1	Joystick -> OSC Gate Joystick 1 Item: Sensor 5...
7	TOT	OSC	GATE	lights	switch 2	Joystick -> OSC Gate Joystick 1 Item: Sensor 6...
8	TOT	OSC	RANG	lights	sub 3 range	Joystick -> OSC Range Joystick 1 Item: Eje X ...
9	AUTO	OSC	TRIG	lights	RESET subs	Automatic -> OSC Trigger: Up End OSC: Lig...

La primera rule cargará automáticamente una memoria de luces (Cue lights) concreta al cargar la parte. Utilizamos para ello como source un trigger, automático, begin (al inicio de cargar esta parte automáticamente)

Como target utilizamos OSC, con el label configurado como Lights Out y en address le escribimos el mensaje de control que DLight reconocerá para ejecutar la memoria de luces que deseamos.

El mensaje es “/seq/X2LoadAndFireCue” y en value ponemos el número de la memoria que queremos cargar, es en este caso la Cue Light número 0.

En el siguiente ejemplo, vamos a cargar la Cue Lights número 1 al activar el sensor 1 del joystick.

Como veis el source también es un trigger, Joystick, sensor 1 (cuando alguien active el sensor número 1)

Como target utilizamos OSC, con el label configurado como Lights Out y en address repetimos el mensaje anterior “/seq/X2LoadAndFireCue” y en value escribimos 1, para que ejecute la Cue lights número 1.

Scene 4 Rule 2 Enabled

Interaction Type: Trigger Sense: Up

Source Type: Joystick Target Type: OSC

Source: Joystick 1 Target: Lights out

Item: Sensor 1 Address: /seq/X2LoadAndFireCue

Value: 1

Actions List

- OSC: Lights out - Address: /seq/X2LoadAndFireCue | Value: 1

Remarks: lights cue 1 Label: lights

OK Cancel

Siguiendo este ejemplo hemos programado controlar con los botones del joystick del 1 al 4 las memorias de luces (Cue lights) de la 1 a la 4.

Scene 4 Rule 3 Enabled

Interaction Type: Trigger Sense: Up

Source Type: Joystick Target Type: OSC

Source: Joystick 1 Target: Lights out

Item: Sensor 2 Address: /seq/X2LoadAndFireCue

Value: 2

Actions List

- OSC: Lights out - Address: /seq/X2LoadAndFireCue | Value: 2

Remarks: lights cue 2 Label: lights

OK Cancel

Scene 4 Rule 4 Enabled

Interaction Type: Trigger Sense: Up

Source Type: Joystick Target Type: OSC

Source: Joystick 1 Target: Lights out

Item: Sensor 3 Address: /seq/X2LoadAndFireCue

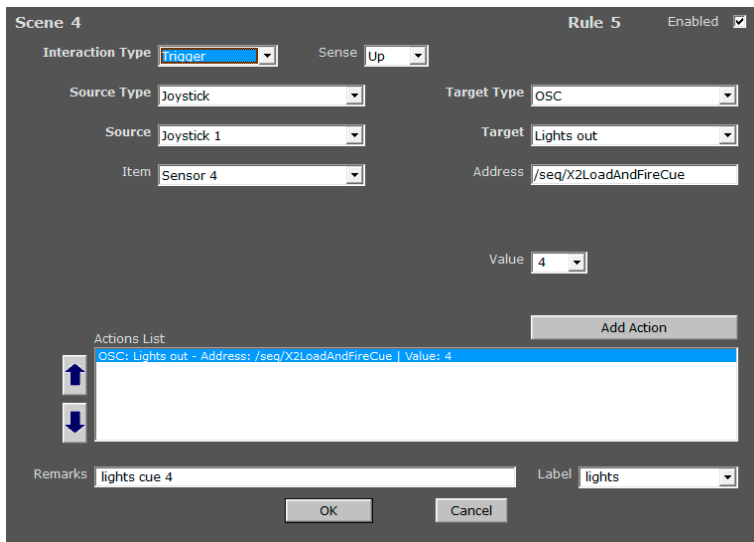
Value: 3

Actions List

- OSC: Lights out - Address: /seq/X2LoadAndFireCue | Value: 3

Remarks: lights cue 3 Label: lights

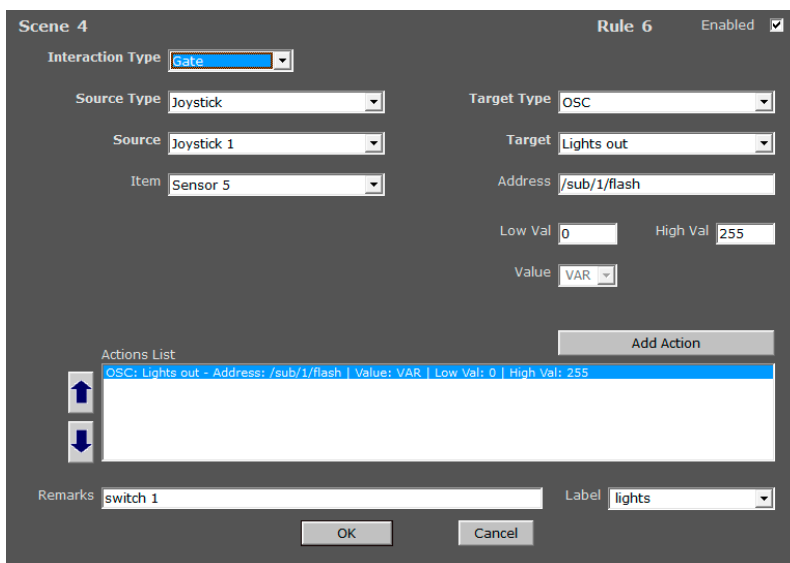
OK Cancel



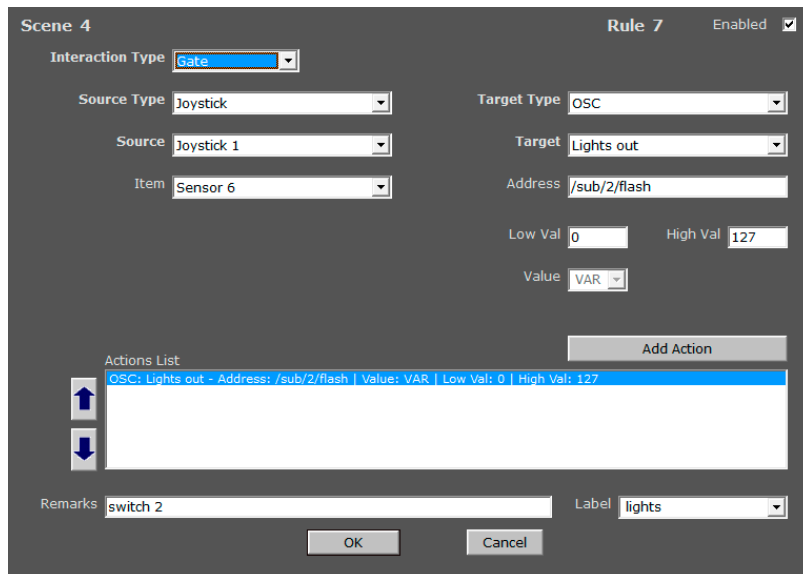
En este ejemplo vamos a programar una rule que ejecute un botón de flash del programa, es decir utilizar un gate para encender un submaster (que puede contener uno o varios focos de luz) al presionar el sensor y que se apague solo al soltarlo. Con esta rule activaremos el submaster 1 en modo flash.

Escojemos un Gate con el source type Joystick y utilizaremos el sensor 5.

Target type OSC, target Lights out y en adress escribimos “/sub/1/flash”. Los valores Low val y High val determinaran que al mandar el mensaje activaremos el flash con el valor 255, es decir, máxima intensidad, 100% del submaster cuando activemos el sensor 5 y 0 indica que lo apagaremos al soltar el sensor 5 a valor 0, es decir al 0%.



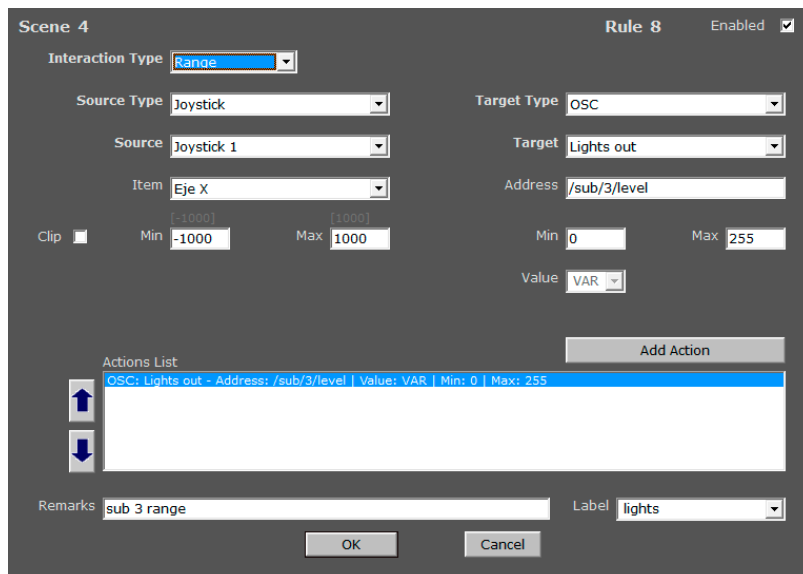
En este segundo ejemplo el sensor 6 será el que activará el submaster 2 en modo flash, todos los items se repiten a excepción que utilizamos el sensor 6 y el adress será “/sub/2/flash” para activar el submaster 2. En este caso también lo hemos variado, pues en el High val hemos escrito el valor de 127 para que el flash se active a la mitad de intensidad 50%, al soltar el botón también se apagará el flash al 0%.



La siguiente rule sirve para controlar con el eje X del joystick el porcentaje de intensidad del submaster 3.

Utilizamos Range, Source type joystick, source joystick 1 y en item, Eje X, dejamos en min y max -1000 y 1000 para aprovechar el máximo rango del joystick.

Target type, seguimos utilizando OSC, target Lights out, adress “/sub/3/level”, min será 0, y máx 255, para activar el submaster a su 100% cuando el joystick esté al máximo de su valor y al 0% cuando el joystick esté al mínimo nivel.



Por último, hemos configurado una rule que resetea los submasters al salir de la parte para evitar que haya quedado activado cualquier submaster y que éstos no queden activos al entrar en una nueva scene.

Trigger, Auto, End en source y en el target utilizaremos OSC, Lights out y como adress escribiremos “/sub/kill” y como valor escribiremos 1.

Scene 4 Rule 9 Enabled

Interaction Type: **Trigger** Sense: **Up**

Source Type: **Automatic** Target Type: **OSC**

When: **End** Target: **Lights out**

Address: **/sub/kill**

Value: **1**

Actions List Add Action

- OSC: Lights out - Address: /sub/kill | Value: 1

Remarks: **RESET subs** Label: **lights**