

# Quartz Composer

## Apuntes

### Interfaz y patches



Quartz Composer es un software propietario de Apple Inc. y parte integrante de las herramientas de desarrollo del sistema operativo Mac OS X.

La propia compañía define a Quartz como una tecnología, como un lenguaje de programación visual y también como un entorno de desarrollo y programación.

Orígenes.

Quartz no fue originalmente desarrollado por Apple sino que proviene de un software de creación visual con aceleración por GPU mediante OpenGL desarrollado por Pierre-Olivier Latour y llamado Pixelshox, el proyecto original todavía se puede consultar bajo esta dirección: [http://www.polhosting.info/web-archives/pixelshox\\_technology/](http://www.polhosting.info/web-archives/pixelshox_technology/)

Por qué es importante QC.

Quartz puede ser usado para muy diversos fines como construir protectores de pantalla, escribir visualizadores RSS, crear nuevos filtros de imágenes para usar en Automator u otro software, crear transiciones de video para Final Cut, efectos para After Effects, etc. Sin embargo, para nosotros la importancia de esta herramienta radica en que los archivos de QC pueden ser ejecutados por diversas aplicaciones de edición de video en tiempo real como pueden ser 2V-P, VDMX, Resolume o Modul8 y desde ahí enviar la imagen a programas de Mapping como Mad Mapper.

Esto no sería una ventaja frente a archivos de video sino fuera por la característica capital de QC: su renderizado en tiempo real.

Así cuando renderizamos un video con After Effects por ejemplo, el archivo resultante queda listo para ejecución y una vez tras otra que lo reproduzcamos obtendremos siempre el mismo resultado. Esto difiere de QC, donde podemos incorporar a nuestra composición elementos como aleatoriedad, interactividad, flujo de datos en tiempo real, etc.

Por lo tanto, si bien para conseguir un look altamente atractivo y profesional como el que obtendríamos después de un renderizado de horas o incluso días desde una aplicación 3D profesional como Cinema4D necesitamos un conocimiento profundo y aplicado de QC, para obtener otros valores como la perfecta sincronía con audio o el tratamiento a tiempo real de la imagen con pocas horas de aprendizaje obtendremos resultados aceptables.

Este es el objetivo de este curso.

## 2.Espacio de Trabajo

### 2.1 Editor

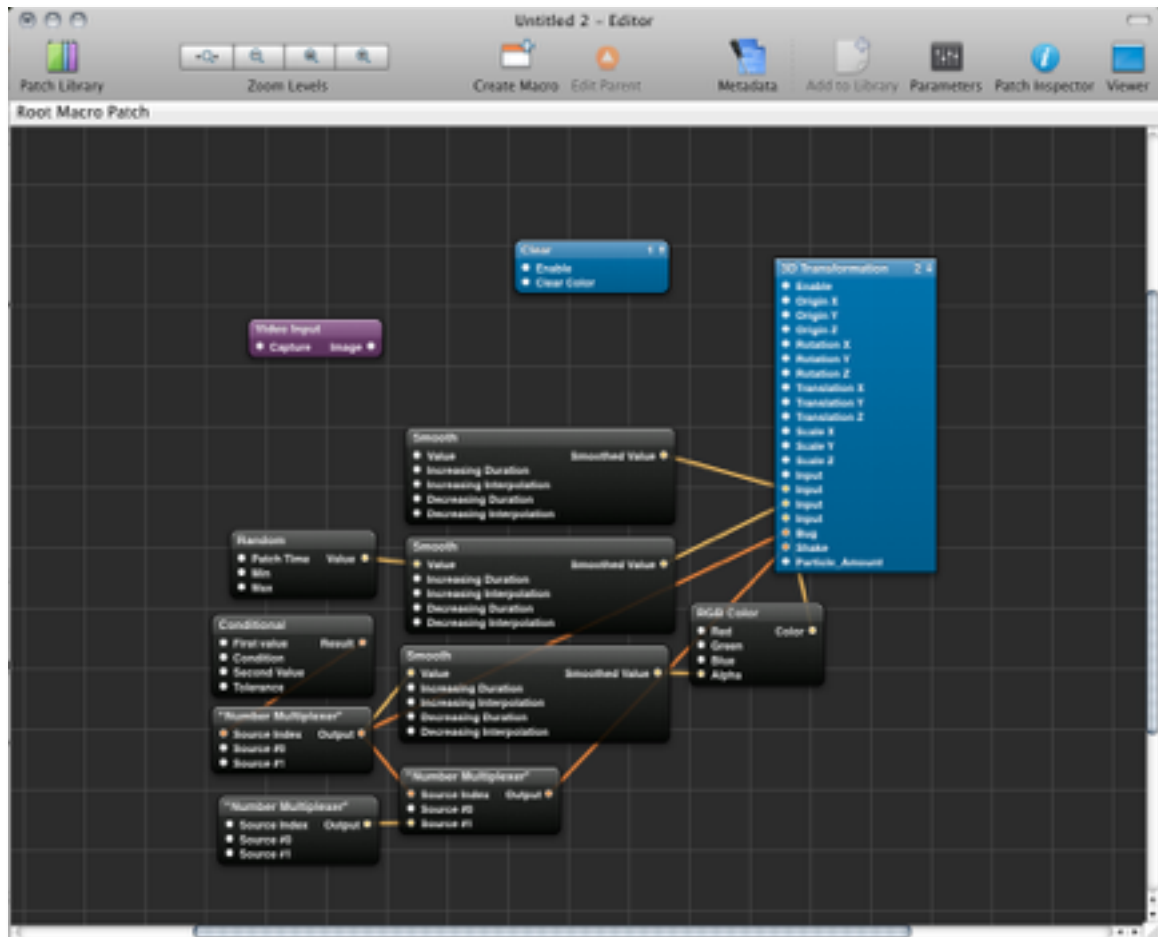
Es el espacio de trabajo, en él se colocan todos los patches que se van a usar y se hacen las conexiones entre ellos.

En la barra de herramientas existen accesos directos al resto de ventanas:

Patch Library

Inspector

Viewer



Existen también controles para el zoom, como elegir el aumento adecuado, mas, menos y volver al original.

Hay un botón para crear un Macro clip y otro para desplazarte al macro superior.

Los macros tienen el borde en ángulo para diferenciarlos de los patches, en la imagen superior se puede apreciar un macro con el nombre 3D Transformation.

Para saltar rápidamente entre los diferentes macros hay una barra de direcciones donde se puede observar la posición actual desde el patch principal o Root Macro Patch.

En la imagen podemos apreciar como el patch GL Line Structure se encuentra dentro de un macro llamado Iterator que a su vez se encuentra dentro de uno llamado Structure, etc. hasta el Root Macro Patch.

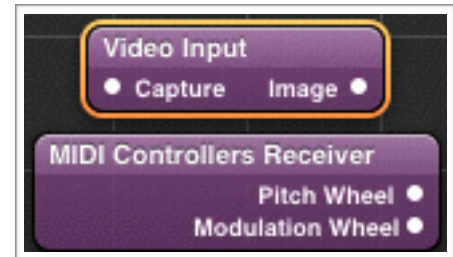
## 1. QC

Quartz Composer está basado en elementos y cables que permiten la conexión entre ellos. Los elementos se conocen como patches y pueden ser 3 tipos (más uno extra): Provider, Processor y Consumer, el tipo extra es una colección ordenada de alguno o todos de los anteriores. Son los Macro Patch y en ellos se basa la jerarquía de patches.

Ejemplos de patches de tipo Provider:

Este tipo de patches poseen en QC 4.0 color magenta.

En la imagen podemos ver un patch para la entrada de video con la típica selección en amarillo y otro patch preparado para la recepción de controles MIDI tipo CC. Los patches de tipo processor son los mas numerosos, se corresponden a las funciones en los lenguajes de programación convencionales.



Los hay de diversos tipos, en este curso veremos los más útiles para nuestro propósito así que se pide a los alumnos que exploren el resto por ellos mismos. Al igual que en cualquier otro aprendizaje de un lenguaje, la gramática es fundamental al igual que el vocabulario básico, el resto de palabras las tendrá que aprender cada cual en base a sus necesidades.

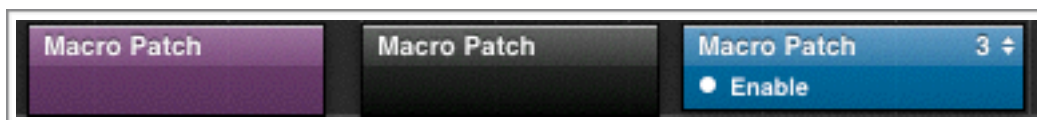
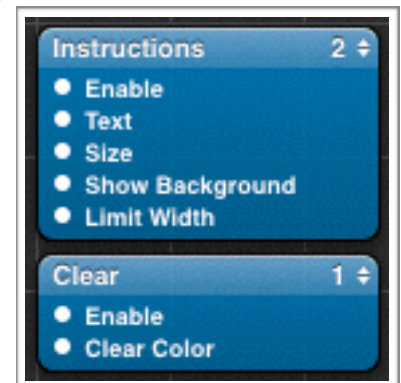
Los patches de tipo Consumer

son los que finalmente ofrecen una salida a las operaciones realizadas por los processors a los providers. En este tipo de tipo se encuentran todos los renderizadores, se caracterizan por poseer color azul además de un número en su esquina superior derecha. Este índice representa el orden de evaluación y renderizado. El índice se puede asimilar al concepto de capas de un software de edición de imagen tradicional, como Photoshop.

Así, en este ejemplo aunque el patch 'Clear' esté situado debajo del 'Instructions' se va a renderizar en primer lugar.

Esta es una diferencia esencial respecto a por ejemplo Max/MSP/Jitter.

El tipo extra es el macro, se distingue del resto ya que posee las esquinas en ángulo a

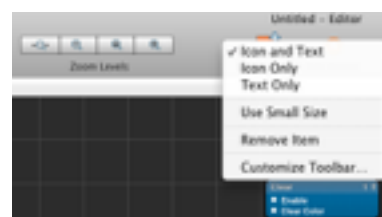


diferencia de las típicamente redondeadas de los demás. Una imagen mental de lo que es un macro podría ser un cajón. Dependiendo de lo que contenga el cajón éste cambiaría su color. Sirven para mantener los proyectos ordenados, para unir variables y para aplicar parámetros de forma conjunta, por ejemplo el macro 3D Transformation, -que es el equivalente a la cámara en QC- ejecuta su transformación a todo lo que se encuentre dentro de él y sólo a eso. Si arrastramos un archivo de QC (extensión .qtz) dentro de otro, el primero aparecerá como un macro dentro del segundo.

Otras diferencias a mencionar respecto a los lenguajes tradicionales son por ejemplo que los loops en Qc se llaman Iterators, las librerías externas se pueden compilar en forma de plugin.

También en la barra de herramientas hay un acceso para añadir a la librería elementos que consideremos de gran utilidad y reusables y un botón que desplegará un panel con los parámetros publicados en el macro actual y en el superior al mismo tiempo.

Estos accesos directos se pueden modificar a placer. Al hacer click con el botón derecho sobre la barra de herramientas aparece este menú:



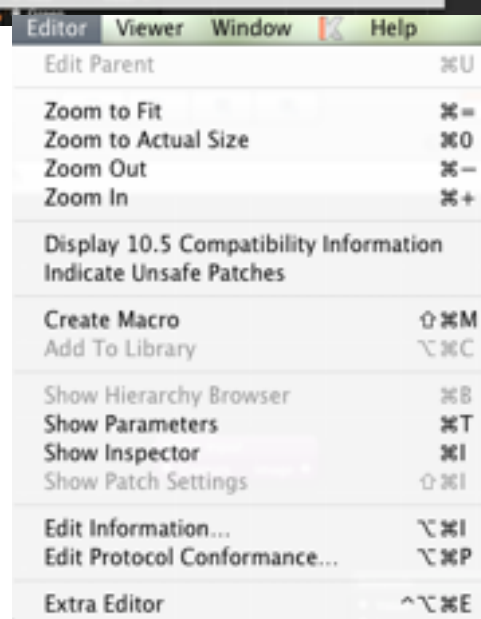
Bajo el submenú Customize Toolbar tenemos acceso a un panel desde el que podremos arrastrar y soltar elementos hacia o desde la barra actual. También se pueden reordenar los iconos pulsando la tecla Cmd y al mismo tiempo haciendo click en uno de ellos.



Un botón importante al que tener acceso es el de Metadata, donde podremos incluir diferentes campos de datos a nuestra composición, por ejemplo copyright, descripción o cualquier atributo que el autor quiera incorporar como por ejemplo lo que veremos más adelante: una mini previsualización para aquellos patches que VDMX no pueda construir por si mismo.

Este hud de metadata es también accesible desde el menú editor y Edit Information... o el acceso rápido de teclado que vemos en la imagen.

En sesiones posteriores veremos con más detenimiento el resto de opciones existentes en la barra de menús.



### 2.3 Patch Library

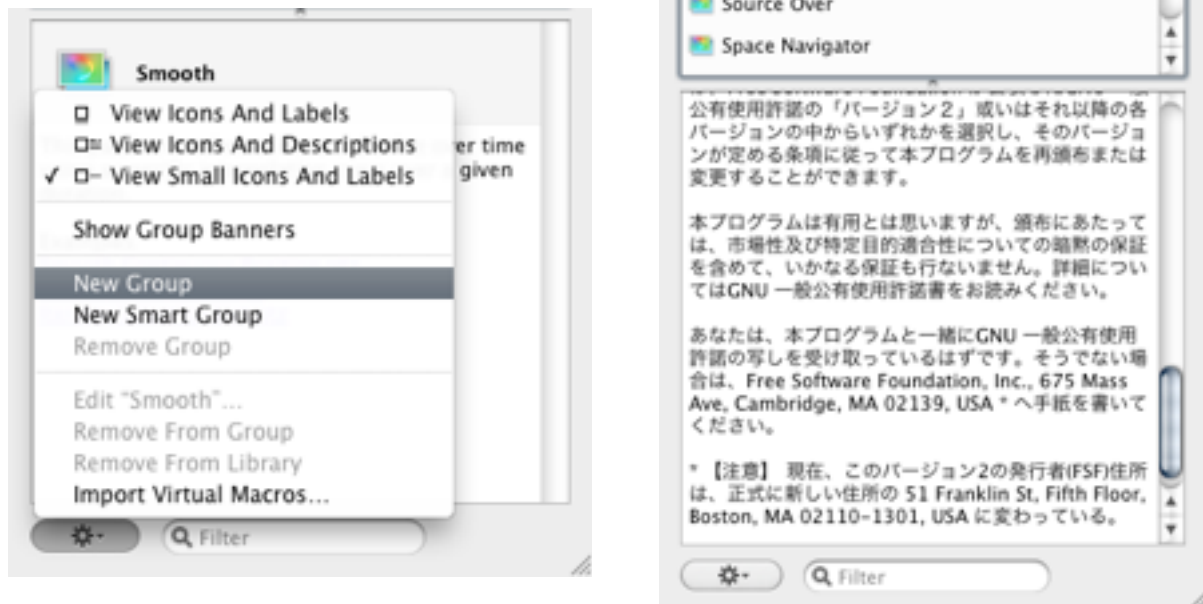
La librería aloja todos los patches que vienen por defecto en QC, además de los que cada usuario haya añadido desde el Editor y también los plugins de terceros que se hayan instalado.

En la imagen se puede apreciar un plugin llamado Single Marker Detector de un desarrollador japonés. Como vemos los patches están ordenados de forma alfabética.

Abajo a la derecha, el botón con el engranaje nos da acceso a las preferencias y en ellas podemos agrupar los resultados según grupo mediante la opción 'Show Group Banners'.

Además, tenemos una función para importar macros desde una ruta específica y las opciones de visualización de los iconos.

Para acceder al panel Library existen varios métodos



como el acceso en el Editor, desde el menú Window/Show Patch Library o pulsando Cmd+Enter con el teclado. Esta última forma es la más rápida ya que coloca el cursor directamente en el cuadro de búsqueda por lo que puedes empezar a teclear el nombre del patch, con las flechas de dirección seleccionar el correcto y al pulsar intro de nuevo colocarlo en el editor. Haciendo doble click sobre el nombre o el icono del patch también lo colocamos en el editor al igual que arrastrándolo y soltándolo.

Resaltar también que podemos filtrar los patches que se muestran dependiendo de su categoría, por ejemplo Library/Image/Gradient. Ahí estarán listados solo Gaussian Gradient, Linear Gradient y Radial Gradient más los patches externos que concuerden con ese filtro.



## 2.2 Viewer.

Es nuestro visor, en él se reflejarán las acciones que repercutan en un patch consumer, esto es debido a que QC usa un método de 'lazy evaluation' por lo que los nodos no se ejecutan hasta que no son llamados a hacerlo, esto es en última instancia hasta que no llegan a un consumer patch.

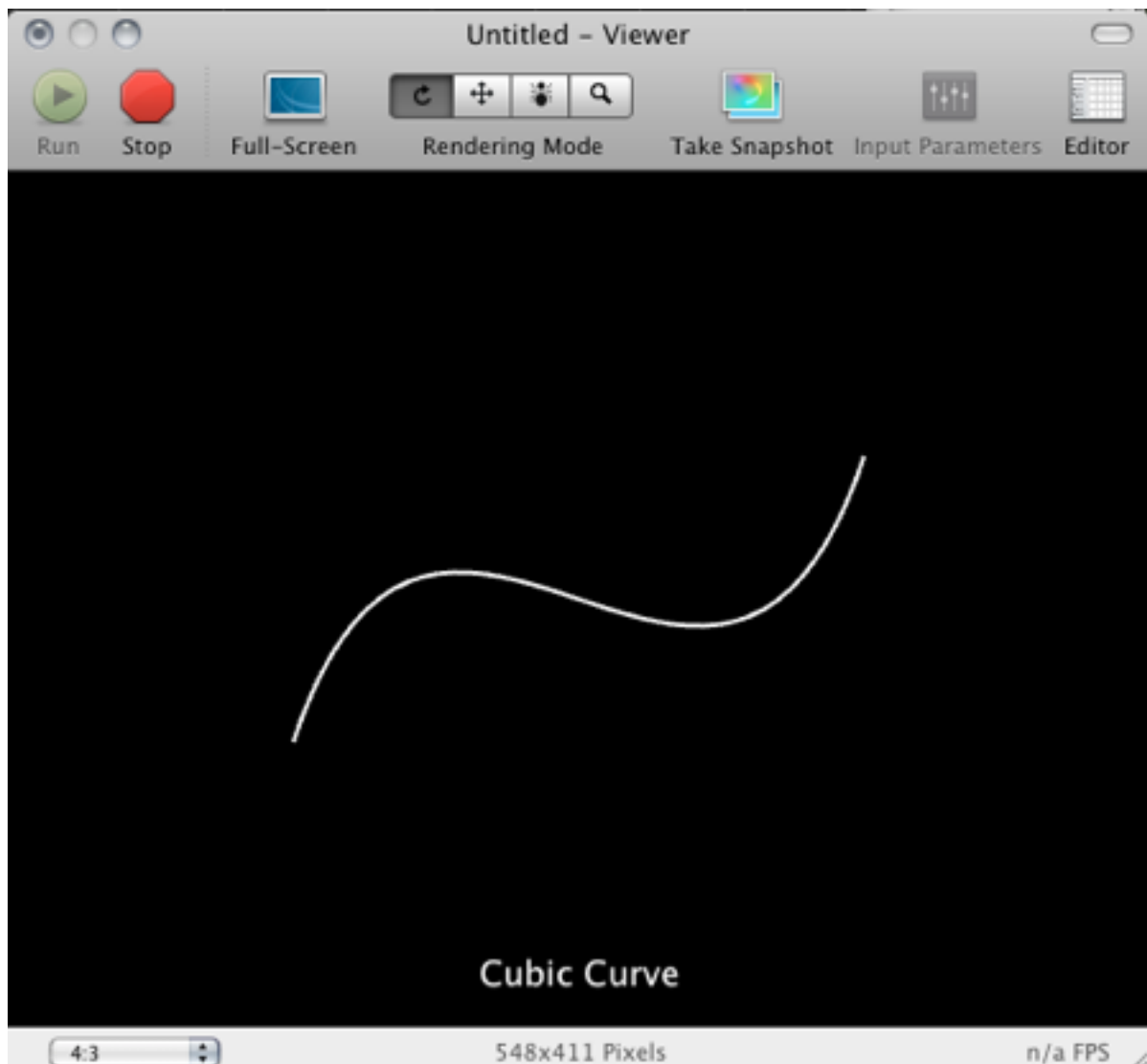
Este es el aspecto general del Viewer:

En la parte superior podemos ver el título del proyecto, en este caso el que QC asigna por defecto, Untitled.

La barra de herramientas es totalmente customizable al igual que la del Editor, en este ejemplo hemos añadido el botón Take Snapshot el cual no está presente por defecto.

El resto de botones son de izquierda a derecha:

Run y Stop para renderizar o no. Fullscreen para llevar el visor a pantalla completa, los diferentes modo de renderizado que veremos más adelante. Un HUD con acceso a los parámetros publicados en el Root Macro Patch (en este caso deshabilitado al no haber publicado todavía ningún parámetro). En última instancia tenemos el acceso al Editor.



En la parte inferior podemos ver el Aspect Ratio (AR), en este caso 4:3 o sea cuatro tercios, los pixeles de ancho y alto, 548x411. Si hacemos un poco de matemáticas al dividir 548 entre 411 obtenemos 1,3 periódico o lo que es lo mismo  $\frac{4}{3}$ . En la esquina inferior derecha podemos observar los Frames Per Second de renderizado, en el ejemplo al ser una imagen estática tenemos un valor de n/a.

El visor de QC posee una diferencia fundamental frente a lo usual en diseño o programación gráfica: el sistema de coordenadas. Los valores asociados a este sistema son relativos.

Así independientemente de la resolución de Píxeles renderizados, el valor de los ejes de ese sistema de coordenadas no varía, esto permite reescalar la imagen final sin deformar la misma en un 90% de ocasiones.

El centro de este sistema de coordenadas se encuentra en el centro de la imagen y corresponde a  $xyz = 0,0,0$ .

El eje X va desde -1 totalmente a la izquierda a 1 en el lado opuesto. De esta forma el ancho de renderizado siempre corresponde a 2 unidades.

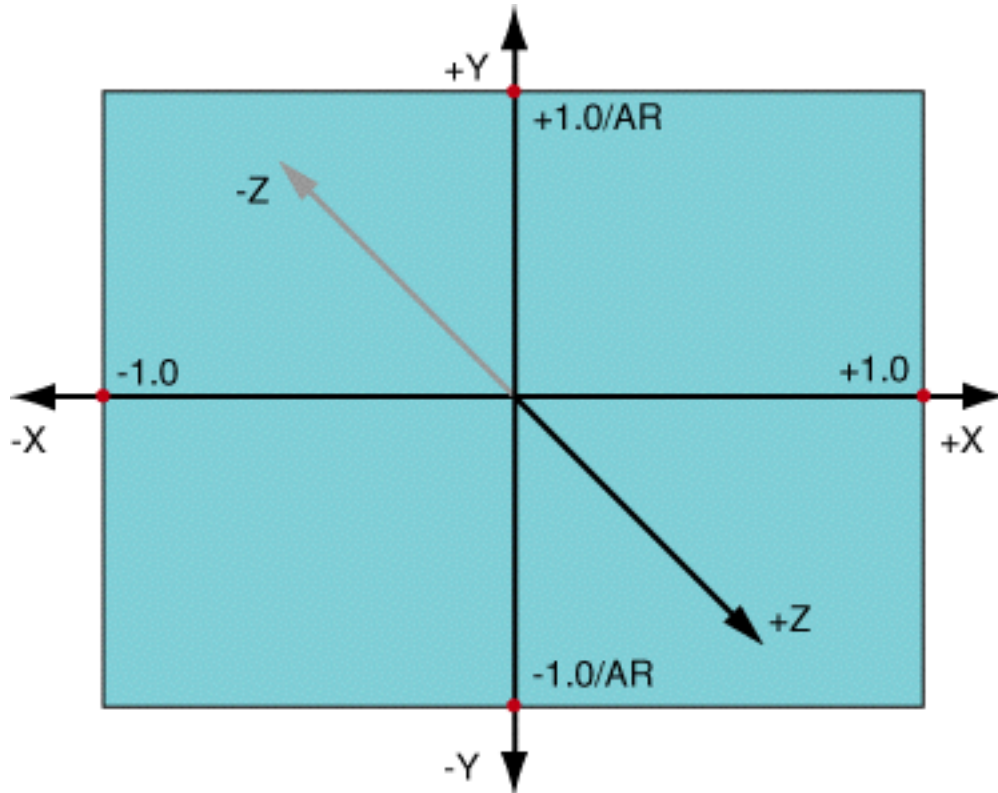
Para el eje Z (profundidad) tenemos valores negativos hacia dentro de la pantalla y positivos hacia fuera.

El eje Y es algo más complicado ya que su punto más alto pertenece al valor  $1/AR$  siendo AR el Aspect Ratio anteriormente comentado. Para un AR de 4:3, el eje Y va desde 0,75 ( $1/(4/3)$ ) en su punto más alto a -0'75 en la parte baja de la pantalla.

Para los Aspect Ratio más comunes tenemos los siguientes valores que corresponden al valor del cruce entre el eje Y y el borde superior de renderizado de la pantalla:

<b>4:3</b>	<b>-&gt; <math>1/(4/3)</math></b>	<b>= <math>1/1,333</math> =</b>	<b>0,75</b>
<b>16:9</b>	<b>-&gt; <math>1/(16/9)</math></b>	<b>= <math>1/1,777</math> =</b>	<b>0,5625</b>
<b>Dv NTSC</b>	<b>-&gt; <math>1/(720/480)</math></b>	<b>=</b>	<b>0,66667</b>
<b>Dv PAL</b>	<b>-&gt; <math>1/(720/576)</math></b>	<b>=</b>	<b>0,8</b>

La siguiente imagen representa el sistema de coordenadas de QC:

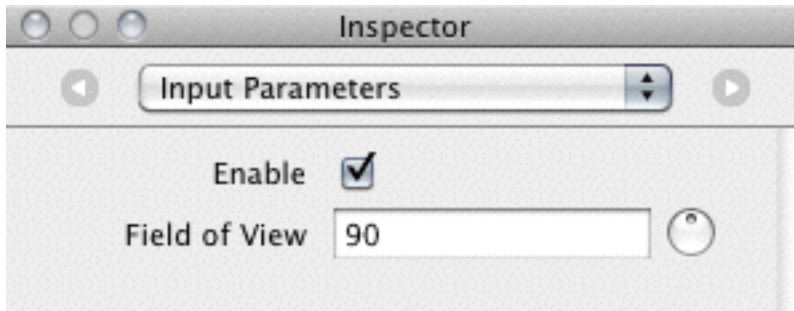






## 2.4 Inspector

El inspector se compone de tres paneles: Input Parameters, Settings y Published Inputs & Outputs. Son accesibles desde Cmd+1, Cmd+2 y Cmd+3 respectivamente.

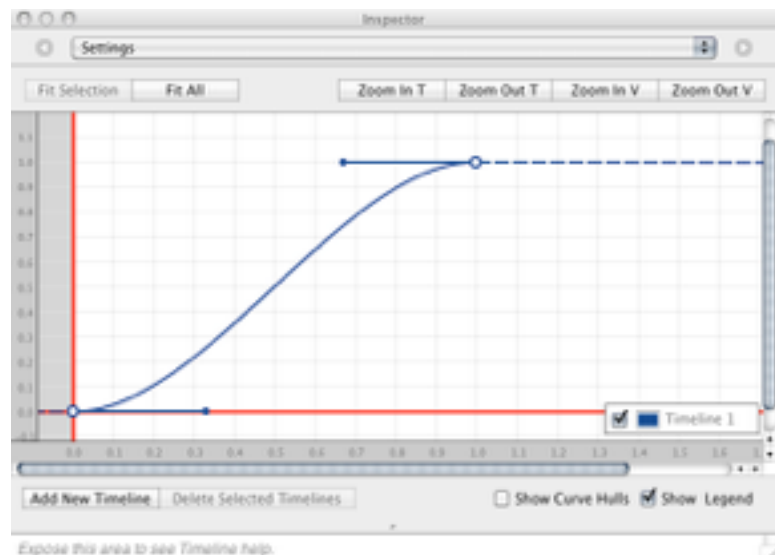


En el primer panel se tiene acceso a las variables del patch seleccionado en el Editor, en este ejemplo el Field of View.

Cada patch tiene unas variables o parámetros diferentes. Además cada parámetro es diferente en si mismo, por ejemplo la translación

en el eje X aceptaría valores positivos y negativos siendo de -1 a 1 los que caerían dentro de pantalla. Para las rotaciones se usan grados, por lo tanto los valores 0, 360 o 720 dan un resultado idéntico y en los rangos intermedios se encontrarían el resto de ángulos.

El panel settings difiere enormemente entre los diferentes patches. Estas preferencias pueden variar el comportamiento de los patches o incluso los parámetros que los rigen. En la imagen se aprecia el panel settings del patch timeline, en el que podemos modificar cómo una variable varía en el tiempo, además podemos añadir más variables lo que repercutirá en el panel de parámetros.



En el panel de Published Inputs & Outputs tenemos una visión general de las variables que se encuentran publicadas para el macro seleccionado o para el Root Macro Patch.

Publicar parámetros.

Al hacer click con el botón secundario sobre un patch accedemos a un menú en el que figuran las opciones publish inputs y publish outputs. Mediante ellas podemos hacer accesibles los parámetros que queramos al macro superior o en el caso del Root los publicamos a la aplicación que actuará de host: 2V-P, VDMX, Coge, Resolume, etc.

Un parámetro publicado tendrá un color diferente al blanco usual y para modificar su valor deberemos ir al inspector del macro superior.



## **2.5 Data Types**

QC trabaja con diferentes tipos de datos.

Los datos presentes en Quartz 4 son los siguientes:

- Boolean, son las que se usan para activar o desactivar patches, los valores posibles son 0 o 1.
- Index, son números enteros que van desde el 0 hasta el 2147483647.
- Number, puede ser un decimal de doble precisión positivo o negativo.
- String, es un campo de texto en formato Unicode.
- Color, pueden ser representados en forma de RGBA, CMYK o Escala de Grises.
- Image, imagen 2D con la posibilidad de que tenga infinita dimensión.
- Structure, un grupo de valores ordenados en forma de árbol.
- Mesh, es un grupo de datos que contienen vértices, per-vertex normals, coordenadas de texturas y colores.
- Virtual, cualquiera de los anteriores.

Se puede transformar directamente entre algunos de esos datos. Por ejemplo, si enviamos un Index a una entrada de tipo Boolean, el 0 nos va a dar “falso” y cualquier número mayor que 0 nos dará “verdadero”.

Es el denominado ‘casting’ en lenguajes de programación tradicionales.

La conexión entre salidas y entradas del mismo tipo de dato se realiza de forma transparente, esto se refleja en un ‘patch cord’ de color amarillo.

Cuando existe una conversión de datos entre la salida y la entrada, por ejemplo desde un dato tipo number a un index, el cable cambiará de color para reflejarlo apareciendo en naranja.

Si el patch cord aparece en rojo quiere decir que una transformación importante se está produciendo y estamos perdiendo la mayor parte de los datos, esto nos puede usar útil en ciertas ocasiones. Por ejemplo, de imagen a boolean se produce esta conversión.

## **2.6 Jerarquía.**

Los patches Macro son contenedores que almacenan patches y reúnen parámetros. Cuando un patch tiene publicado un parámetro cede de esta forma su control al patch superior, en su defecto al Root Macro Patch.

En esta imagen observamos como para el patch Line Structure tenemos publicado el parámetro Line Width, por lo que como vemos en el Inspector, ya no podemos alterar su valor. En la barra de direcciones vemos que deberíamos desplazarnos al Root Macro Patch para acceder a ese valor.

Un Splitter se puede crear para cualquier parámetro desde el menú desplegable a través del botón secundario. Gracias a los splitters podemos marcar un rango para los parámetros que se usan en 2V-P o VDMX. Para ello usamos el panel settings de la ventana Inspector. Además los splitters nos permiten usar el mismo valor para dos entradas diferentes y por otro lado mantener los outputs publicados dentro de un Macro modificando el flujo interno del patch sin hacerlo en el exterior.

Otra función importante de los macros es la de aplicar a todo su interior un comportamiento específico.

Desde Librería/Environment tenemos acceso a los instalados en el sistema.

Por ejemplo Lighting aplica una o varias luces a todo lo que esté contenido dentro de el. 3D Transformation nos permite escalar, rotar y desplazar los elementos contenidos en su interior de una forma global. Es el equivalente en QC a una cámara.

Para temas de cálculo de posiciones y perspectiva es bueno conocer el valor del punto de vista que posee el viewer, éste está colocado en el punto de coordenadas  $xyz=(0,0,1'73205)$ .

El Macro Environment 'Render in Image' nos da la posibilidad de convertir cualquier patch consumer en una entrada para otro patch consumer. Debemos activar dependiendo de la situación sus preferencias en el panel correspondiente además de limpiar el buffer con el patch clean en su interior.