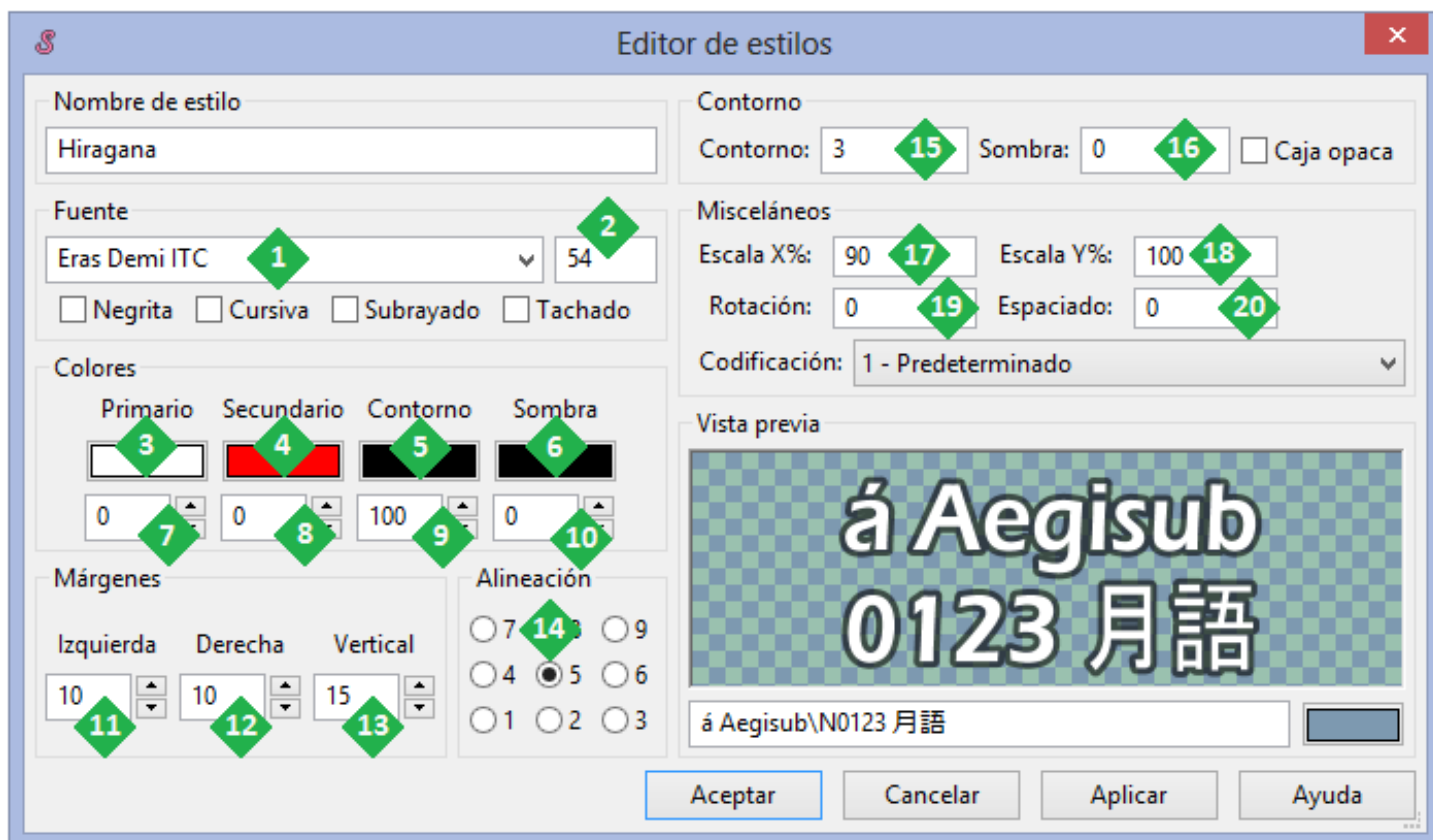

Kara Effector 3.2:

Y ha llegado la hora de empezar a ver las Librerías con las que cuenta en **Kara Effector 3.2** y para ello he dispuesto una serie de ejemplos y listados para tratar que cada una de ellas quede lo más clara posible. El conocimiento y dominio de las Librerías es indispensable a la hora de hacer y modificar Efectos de alta calidad.

Librería "I":

Es un listado de variables y constantes que hacen referencia a valores de la "Línea", es decir, a cada una de las líneas del archivo .ass.

La mayoría de los valores de esta Librería son los que asignamos en el Estilo de Línea en el editor de estilos del **Aegisub**:



1. l.fontname
2. l.fontsize
3. l.color1
4. l.color2
5. l.color3
6. l.color4
7. l.alpha1
8. l.alpha2
9. l.alpha3
10. l.alpha4
11. l.margin_l
12. l.margin_r
13. l.margin_v, l.margin_t, l.margin_b
14. l.align
15. l.outline
16. l.shadow
17. l.scale_x
18. l.scale_y
19. l.angle
20. l.spacing

Los anteriores 20 valores de la Librería “l” no necesitan mucha explicación, ya que son los que siempre usamos cuando creamos y modificamos un nuevo Estilo de Línea.

Un pequeño ejemplo sería:

```
Add Tags:          Add Tags Language: Lua
string.format("\t\t(0,200,\fscx200)\t\t(200,1000,\fscx%s)", l.scale_x)
```

En el ejemplo, se aumenta la escala en el eje “x” en 200% desde 0 hasta 200 ms y luego desde 200 ms a 1000 ms la escala en el eje “x” vuelve a la proporción que tiene en el Estilo, es decir, 90% (como se puede apreciar en la imagen anterior del Estilo, ítem 17).

Veamos este mismo ejemplo, pero ahora en lenguaje **Automation Auto-4**. Aunque la forma de escribirlo cambie dado los dos tipos de lenguajes, el resultado es el mismo:

```
Add Tags:          Add Tags Language: Automation Auto-4
\t\t(0,200,\fscx200)\t\t(200,1000,\fscx!l.scale_x!)
```

Y continuando con los valores de la Librería “l”, ahora veremos los que son referentes a las Líneas de Texto y a su posición en el vídeo.

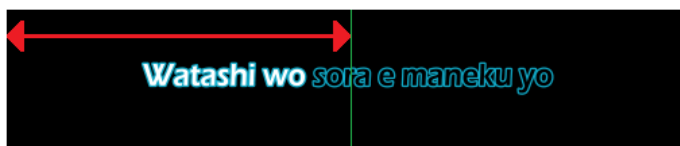
l.width: es el Ancho medido en pixeles de cada una de las Líneas.



l.left: es la Distancia medida en pixeles desde la parte izquierda del vídeo hasta la parte izquierda de la Línea.



l.center: es la Distancia medida en pixeles desde la parte izquierda del vídeo hasta el centro de la Línea.



l.right: es la Distancia medida en pixeles desde la parte izquierda del vídeo hasta la parte derecha de la Línea.



l.height: es la Altura medida en pixeles de cada una de las Líneas.



l.top: es la Distancia medida en pixeles desde la parte superior del vídeo hasta la parte superior de la Línea.



Notarán que la parte superior de la Línea no coincide con la parte superior de sus letras más grandes, esto se debe a que hay un espacio extra que sirve para separar las Líneas verticalmente cuando estas están una encima de la otra. Ya que de otro modo la Líneas se verían pegadas.

l.middle: es la Distancia medida en pixeles desde la parte superior del vídeo hasta la mitad de la altura de la Línea.



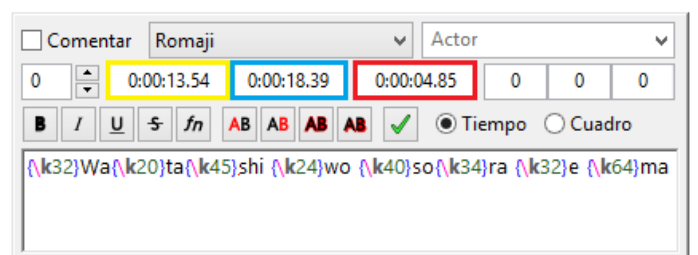
l.bottom: es la Distancia medida en pixeles desde la parte superior del vídeo hasta la parte inferior de la Línea.



l.descent: es la Distancia medida en pixeles desde la parte superior de la Línea hasta la parte superior Real de la misma. Es la Distancia que separa verticalmente una Línea de otra.



Para terminar, los valores de la Librería “l” que hacen falta son los que tienen referencia con el tiempo y su forma de escritura:

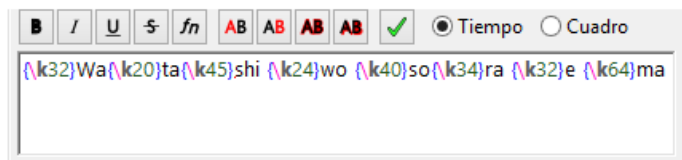


l.start_time: (recuadro amarillo de la anterior imagen) es el tiempo de inicio medido en ms (milisegundos) de cada Línea.

l.end_time: (recuadro azul) es el tiempo final medido en (ms) de cada Línea.

l.duration: (recuadro rojo) es la duración total medida en (ms) de cada Línea de Dialogo del archivo .ass.

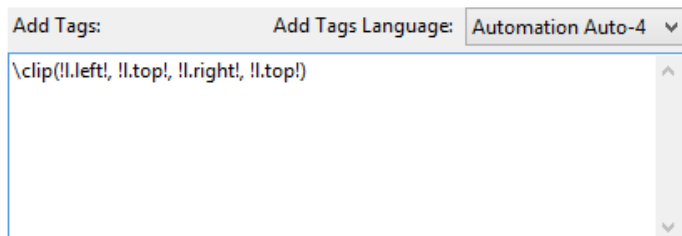
l.text: es todo lo que está escrito en la Línea, incluidos los tags:



l.text_stripped: es similar a **l.text**, pero con la diferencia que no tiene en cuenta a los tags que tenga dicha Línea:

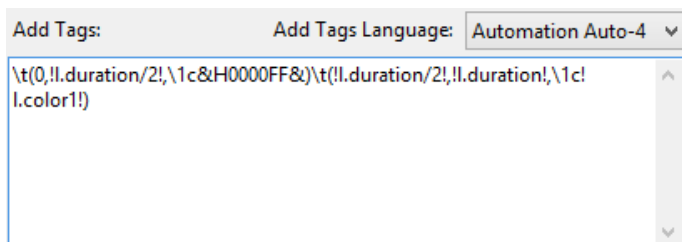


El uso de cada uno de los valores de la Librería “l” o de las próximas librerías que veremos, facilitará la forma de hacer Efectos que en principio serían mucho más complejos. Acá otro corto ejemplo de cómo poder usar la Librería “l”:



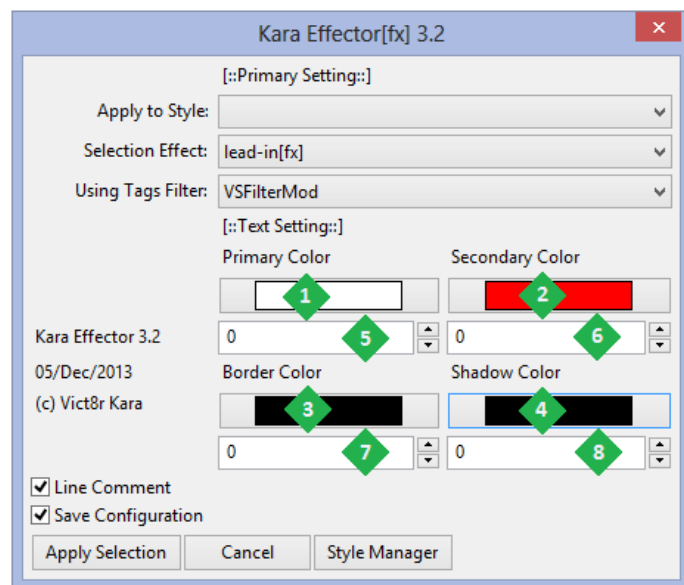
Es un clip en donde es visible toda la Línea.

En este otro ejemplo usamos **l.duration** para generar una transformación del color primario. Desde 0 hasta la mitad de la duración, el color primario se transforma en Rojo (**&H0000FF&**: Rojo en formato .ass) y desde la mitad de la duración hasta la duración total, regresa a su color original asignado en el Estilo (**l.color1**):

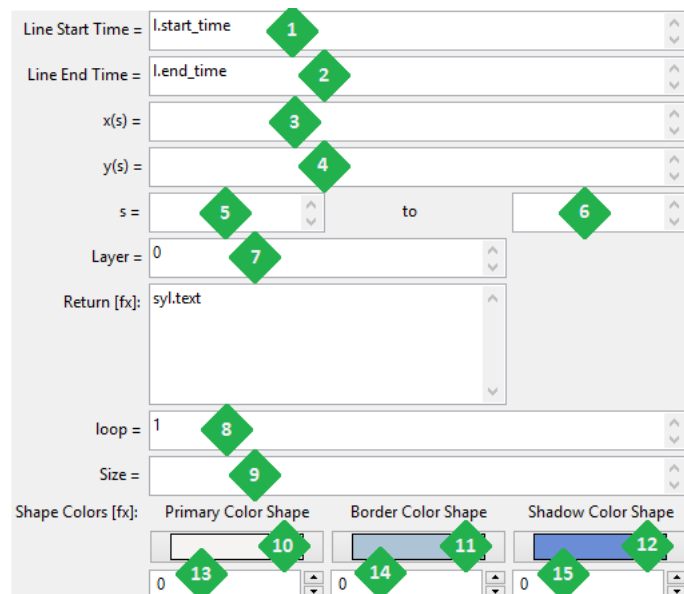


Librería “fx”:

Es la Librería que contiene los valores del Kara Effector.



1. **text.color1**
2. **text.color2**
3. **text.color3**
4. **text.color4**
5. **text.alpha1**
6. **text.alpha2**
7. **text.alpha3**
8. **text.alpha4**



1. **fx.start_time**

2. **fx.end_time**
3. **fx.fun_x**
4. **fx.fun_y**
5. **fx.domain_i**
6. **fx.domain_f**
7. **fx.layer**
8. **fx.maxloop_fx**
9. **fx.size_x, fx.size_y**
10. **shape.color1**
11. **shape.color3**
12. **shape.color4**
13. **shape.alpha1**
14. **shape.alpha3**
15. **shape.alpha4**

16. **fx.center_x**
17. **fx.center_y**
18. **fx.scale_x**
19. **fx.scale_y**
20. **fx.align**
21. **fx.move_x1, fx.move_x2, fx.move_x3, fx.move_x4**
22. **fx.move_y1, fx.move_y2, fx.move_y3, fx.move_y4**
23. **fx.movement_i, fx.movement_f**

No parecen ser muchas, pero son suficientes. Ahora veremos una pequeña explicación de cada una de ellas.

Bueno, considero que los valores de la ventana de inicio del **Kara Effector** no necesitan de mucha explicación ya que son los colores y transparencias que tendrán por default cada una de las Líneas de Efecto generadas.

fx.start_time: es el tiempo de inicio de cada una de las Líneas generadas por un Efecto. Puede ser modificado directamente desde su respectiva celda de texto o con la función **retime** desde “Add Tags” o alguna función hecha en “Variables”.

fx.end_time: es el tiempo final de cada una de las Líneas generadas por un Efecto. Este valor puede ser modificado directamente desde su respectiva celda de texto o con la función **retime** desde “Add Tags” o alguna función hecha en “Variables”. La variable **fx.start_time** puede ser usada en esta celda de texto, ejemplo:

O sea que **fx.start_time** es un valor aleatorio entre -2000 ms y 2000 ms respecto al tiempo de inicio original de cada Línea seleccionada para aplicarle un Efecto (**l.start_time**).

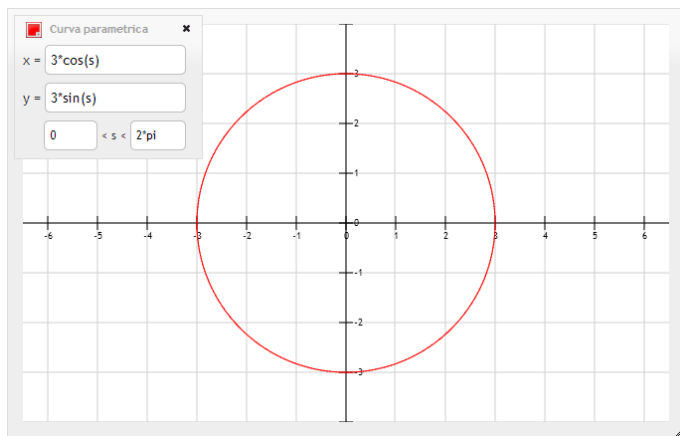
Por otro lado, **fx.end_time** pone que será igual al tiempo en donde inició la Línea de fx (**fx.start_time**), sumado a 500 ms. Es fácil deducir que el **fx.dur** de todas las Líneas que se generen con estos dos tiempos, siempre será de 500 ms. (véase la siguiente variable).

fx.dur: es la duración total de cada Línea fx que sea generada por un Efecto. Es equivalente a la siguiente diferencia:

$$\text{fx.dur} = \text{fx.start_time} - \text{fx.end_time}$$

fx.fun_x: es la ecuación paramétrica de “x” en términos de “s” (donde “s” es el dominio de la función). Ejemplo:

En la anterior imagen están las ecuaciones paramétricas de “x” y “y” de un Círculo de Radio 3. También podemos ver el dominio “s” que va desde 0 a 2π :



La anterior gráfica fue generada en <http://fooplot.com>

En el Kara Effector pondríamos así:

$x(s) = 3 \cdot \cos(s)$
 $y(s) = 3 \cdot \sin(s)$
 $s = 0$ to 2π

Las escalas en el **Kara Effector**, tanto en el eje “x” como en el “y” son por default 1, es decir que este círculo se verá en el vídeo muy pequeño, ya que el Radio del Círculo es solo de 3 pixeles. Hay dos formas de aumentar su tamaño: o se aumenta el Radio del Círculo directamente en las ecuaciones paramétricas o se aumentan las escalas en las celdas de texto que están al lado derecho de estas, todo depende del gusto y del nivel de habilidad adquirido de cada uno.

En estas celdas de texto podemos modificar las Escalas en ambos ejes de una determinada gráfica:

Scale in 'X' =
 Scale in 'Y' =

Olvidaba mencionar que este tipo de Efecto se hacen con Shapes y con un **loop** lo suficientemente grande para que se generen la gráficas.

Para un ejemplo guiado en el **Kara Effector** tomando como base las ecuaciones paramétricas del Círculo, sería:

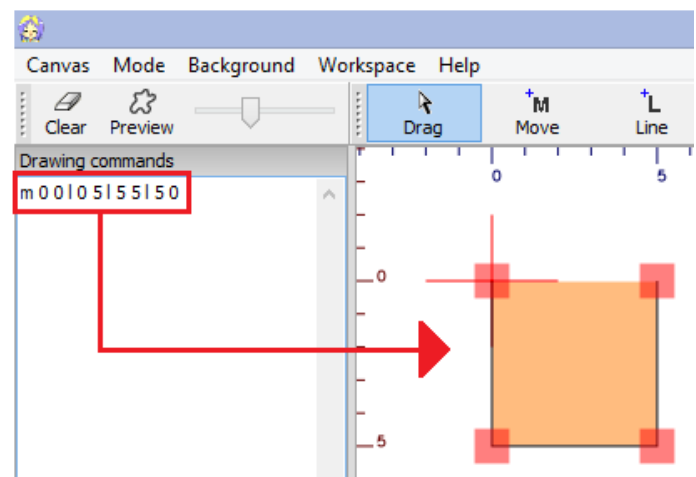
A. Definimos las ecuaciones paramétricas y el dominio de las funciones (“s”):

$x(s) = 3 \cdot \cos(s)$
 $y(s) = 3 \cdot \sin(s)$
 $s = 0$ to 2π

B. Aumentamos las escalas en 10 para que el Radio del Círculo pase de 3 pixeles a $3 \cdot 10 = 30$ pixeles. (Se puede experimentar con distintos valores en las escalas y ver cómo varían las gráficas):

Scale in 'X' = 10
 Scale in 'Y' = 10

C. Elegimos una **Shape** y la ponemos en **Return [fx]**:



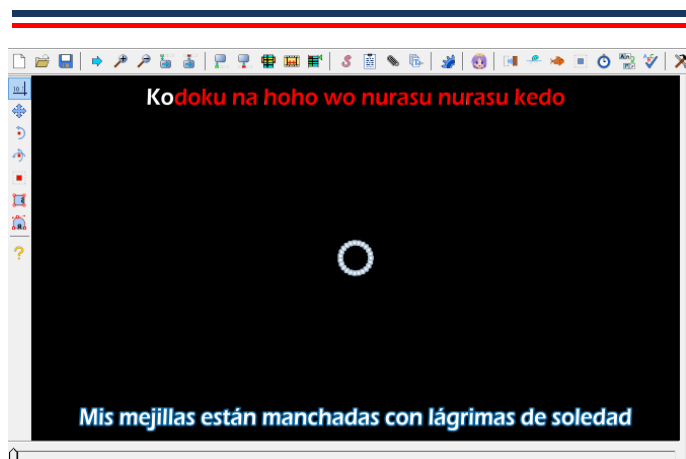
En mi caso, dibujé un Cuadrado en el **AssDraw3**, de 5 pixeles de lado y copié el código de la Shape en **Return [fx]**:

Return [fx]: m 0 0 1 0 5 1 5 5 1 5 0

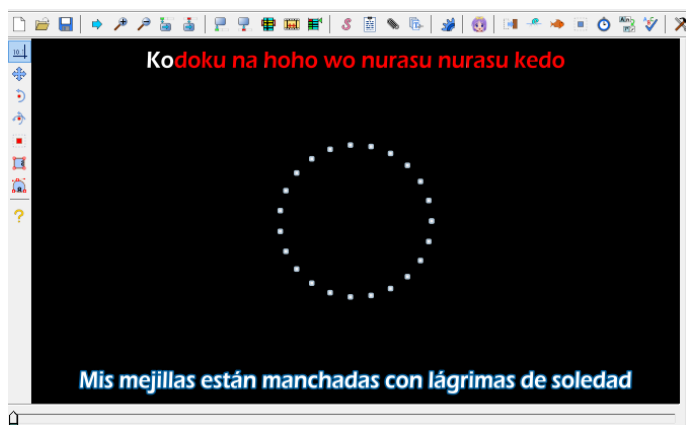
D. Aumentamos el **loop** para que generará la gráfica:

loop = 24

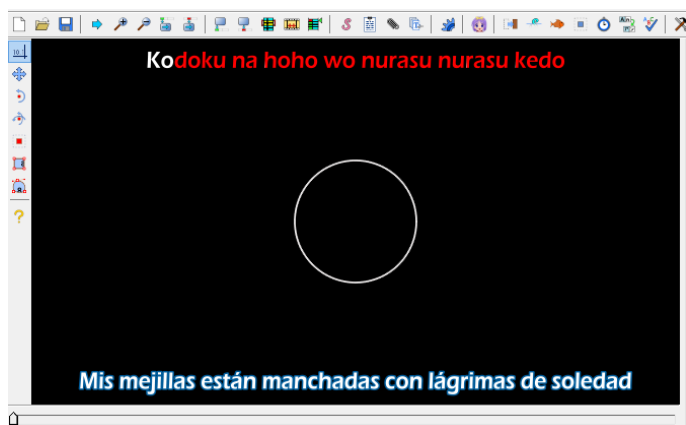
Para este ejemplo puse en **Template Type** la opción “Line” para que se genere un solo Círculo por cada Línea:



Ya se puede ver en el vídeo el Círculo que se generó, pero parece que 30 pixeles para el Radio es poco para poder ver los 24 cuadraditos (loop) de 5 X 5 pixeles que lo conforman, por ello aumentaré las escalas en ambos ejes a 50 y volveré a aplicar el Efecto:



Ahora sí se pueden apreciar los cuadraditos que generan el Círculo. La cantidad de cuadrados, lo mismo que el valor de las escalas, tamaños (size) y colores, se pueden modificar para que la gráfica sea más continua, ejemplo:



Es un ejemplo de cómo “graficar” en el **Kara Effector**.

Las configuraciones para generar el anterior Círculo son:

lead-in[fx]: ABC Template

Line Start Time =

Line End Time =

x(s) =

y(s) =

s = to

Layer =

Return [fx]:

Pos in 'X' =

Pos in 'Y' =

Times Move =

loop =

Size =

Shape Colors [fx]: Primary Color Shape Border Color Shape Shadow Color Shape

Template Type [fx]:

Center in 'X' =

Center in 'Y' =

Scale in 'X' =

Scale in 'Y' =

Align [\an] =

Pos in 'X' =

Pos in 'Y' =

En próximos ejemplos veremos cómo hacer un Efecto a partir de la gráfica de las ecuaciones paramétricas.

fx.fun_y: es la ecuación paramétrica de “y” en términos de “s” (donde “s” es el dominio de la función). Ejemplo:

Curva parametrica

x =

y =

< s <

Con el ejemplo anterior del Círculo, esta variable de la Librería “fx” no necesita muchas más explicación y por ello iremos directamente a las siguientes variables.

fx.domain_i: es el inicio del dominio “s” de las ecuaciones paramétricas. Si en esta celda de texto no pone nada, el inicio del dominio por default es 0.

fx.domain_f: es el final del dominio “s” de las ecuaciones paramétricas. Si en esta celda de texto no pone nada, el final del dominio por default es 1.

No necesariamente el inicio del dominio “s” debe ser menor que el final, lo convencional es que sí lo sea para que la gráfica se dibuje normalmente de menor a mayor, de lo contrario de dibujará al revés. Este orden cobra importancia cuando hacemos animaciones con las gráficas y las vemos en el vídeo.

fx.layer: es la capa de cada una de las Líneas de fx y decide la prioridad en el vídeo de dos o más Objetos Karaoke que coinciden de forma total o parcial en su posición o trayectoria de movimiento.

fx.maxloop_fx: este valor equivale a la cantidad total de repeticiones de cada una de las Líneas fx. Se puede usar con este nombre o con uno más conocido por aquellos que ya conocen algo de **Automation Auto-4: maxj**

$$\text{maxj} = \text{fx.maxloop_fx}$$

El valor de **maxj** (Máximo valor de “j”) se puede modificar directamente en la celda de texto destinada para ello o con la función **maxloop**.

En el caso de haber un solo valor en esta celda, el valor de **maxj** es ese mismo valor:

En este caso, **maxj** = 2

Para el caso de dos valores:

Para este caso, **maxj** = 2 x 5 = 10

Y para el caso de que haya tres valores:

Y para este último caso, **maxj** = 2 x 5 x 4 = 40

3 es el número máximo de valores que se pueden poner en esta celda de texto. Cuando alguno de ellos o todos, no están, su valor por default es 1.

Más adelante veremos por qué la necesidad de más de un valor en esta celda de texto, pero por ahora les mostraré la variable asignada a cada uno de ellos, ejemplo:

fx.loop_v: (el 2 en la imagen) es el loop vertical.

fx.loop_h: (el 5 en la imagen) es el loop horizontal.

fx.loop_3: (el 4 en la imagen) es un loop de respaldo.

Como lo mencionaba antes, si alguno de estos tres valores no está, por default es 1. Lo que quedaría:

$$\text{fx.maxloop_fx} = \text{fx.loop_v} * \text{fx.loop_h} * \text{fx.loop_3}$$

$$\text{maxj} = \text{fx.loop_v} * \text{fx.loop_h} * \text{fx.loop_3}$$

$$\text{maxj} = \text{fx.maxloop_fx}$$

fx.size_x: es el porcentaje del tamaño con respecto al eje “x” asociado al tag \fscx, ejemplo:

En este ejemplo, **fx.size_x** valdría 80 y en el Efecto se verán los siguientes tags: \fscx80\fscy80

Es decir que si solo hay un valor en esta celda de texto, el porcentaje en el eje “y” será el mismo que para el eje “x”, o sea 80% para los dos ejes.

fx.size_y: es el porcentaje del tamaño con respecto al eje “y” asociado al tag \fscy, ejemplo:

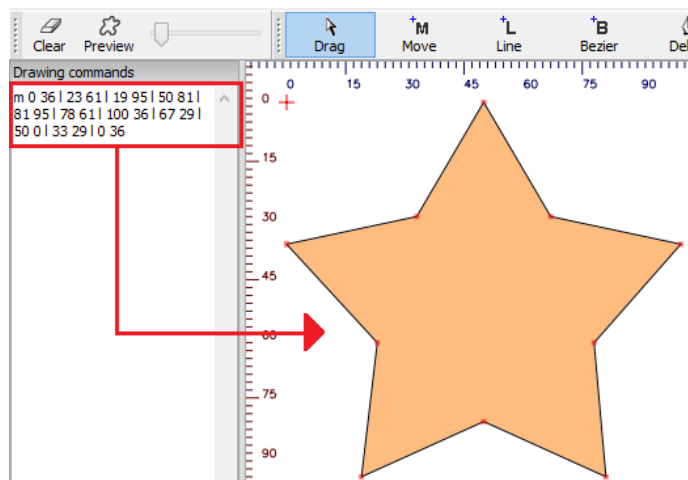
O sea: **fx.size_x** = 80 y **fx.size_y** = 125

Y en tags: \fscx80\fscy125

Si esta celda de texto está vacía los valores por default de estas dos variables son:

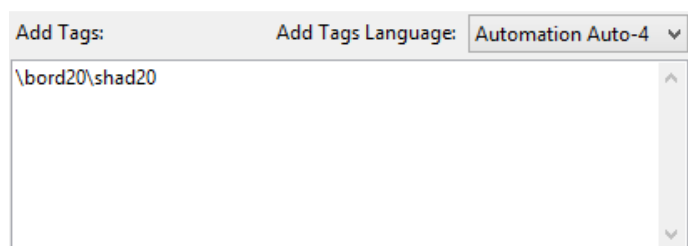
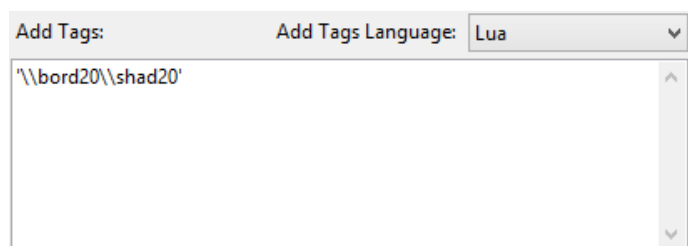
$$\text{fx.size_x} = \text{l.scale_x} \text{ y } \text{fx.size_y} = \text{l.scale_y}$$

Para las variables de la Librería “fx” concernientes a las figuras hechas en el **AssDraw3** usaré la siguiente Shape:



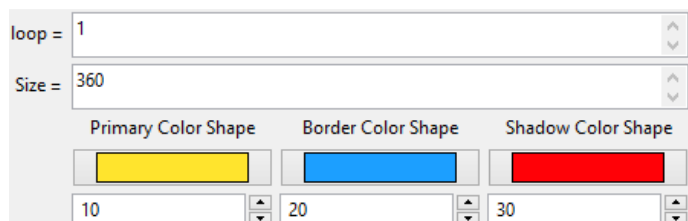
Return [fx]: m 0 36 | 23 61 | 19 95 | 50 81 | 81 95 | 78 61 | 100 36 | 67 29 | 50 0 | 33 29 | 0 36

Y le agregaré los siguientes tags:

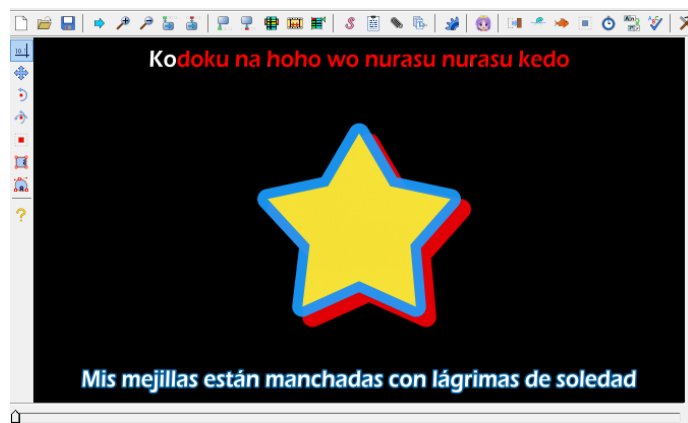


O sea, 20 pixeles para el tamaño del borde y otros 20 para el tamaño de la sombra.

Además de esto, las siguientes configuraciones:



Y al aplicar el Efecto, se verá en pantalla la Shape con las configuraciones que le di:



shape.color1: es el color Primario de la Shape (amarillo).

shape.color3: es el color del Borde de la Shape (azul).

shape.color4: es el color de la Sombra de la Shape (rojo).

shape.alpha1: es la transparencia del color Primario de la Shape (10 para este ejemplo).

shape.alpha3: es la transparencia del color del Borde de la Shape (20).

shape.alpha4: es la transparencia del color de la Sombra de la Shape (30).

Para el **Tomo IV** terminaremos con la explicación de las variables restantes de la Librería “fx” y veremos más de las librerías del **Kara Effector**. No olviden visitarnos en el **Blog Oficial** lo mismo que en los canales de **YouTube** para descargar los nuevos Efectos o si quieren comentar, exponer alguna duda o hacer alguna sugerencia.